1994年1月1日発行（毎月1回1日発行）第13巻1号通巻141号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

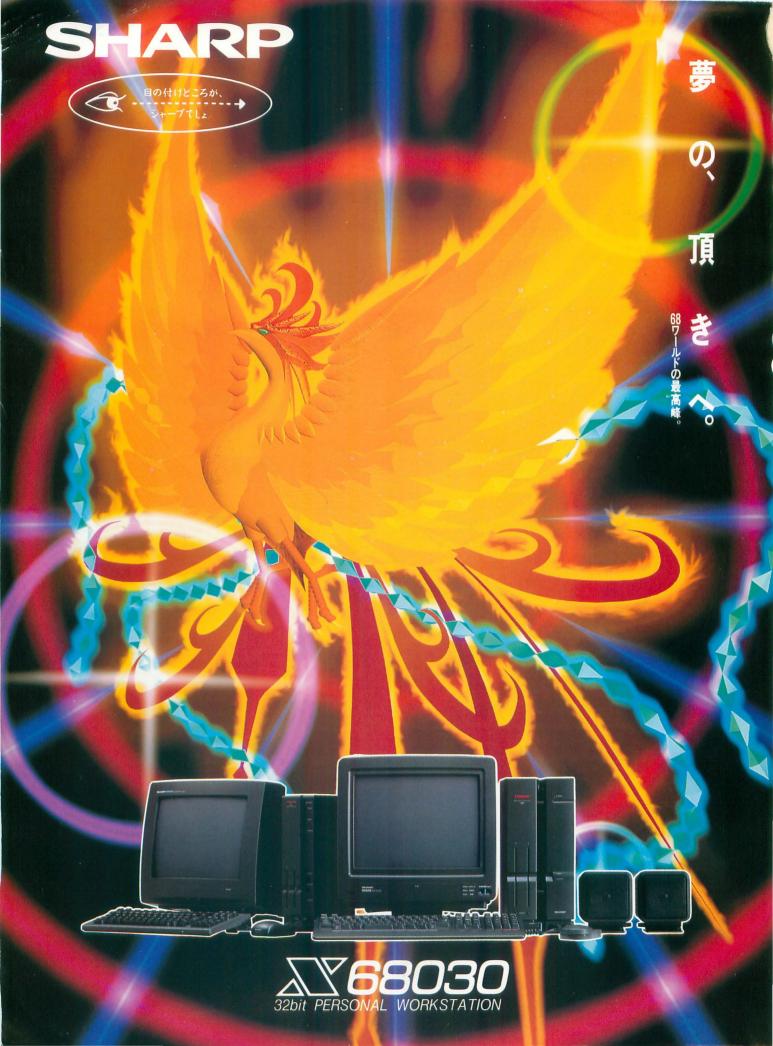
Oh!X

特集 Z-MUSICシステムver.2.0
新機能の概要/ZPCNV.R/トラックワークの使い方
シミュレーションゲーム「NUCLEAR WAR」
新連載"実戦!"ゲーム作りのKNOW HOW

1
1994

SOFT BANK オー/エックス
定価600円

SHARP
目の付けどころが、
シャープでしょ。
夢の、頂きへ。
68ワールドの最高峰。
X68030
32bit PERSONAL WORKSTATION

32ビットパーソナルワークステーション

演算速度4.3倍(当社10MHz機比)/2.4倍(当社XVI比)※1、動画ウィンドウに見る新創造次元。選ばれた人だけが持つ感性によってX68030の扉はひらかれる。

X68000シリーズとして初の32ビットMPU MC68EC030を搭載して高速化を実現。

データキャッシュ、プログラムキャッシュをそれぞれ256バイト搭載したクロック周波数25MHzの高速32ビットMPUを搭載。演算速度は2倍以上(当社従来比)※1の高速化を実現しました。また数値演算プロセッサMC68882※2(25MHz)もサポート。大量の実数演算を必要とするクリエイティブワークやGUI環境の操作性など、実行速度の飛躍的な向上が図られています。(当社従来比)

※1 Dhrystn(四則演算)比。25MHz・データキャッシュオン・プログラムキャッシュオンでMC68000/10MHz時の約4.3倍、16MHz時の約2.4倍。

※2 数値演算プロセッサCZ-5MP1標準価格54,800円(税別):本体内の専用ソケットに取りつけ可能。

65,536色表示、動画表示を実現。さらにパワーアップしたSX-WINDOWver.3.0。

X68000独自のウィンドウシステムとして定評の「SX-WINDOWver.2.0」をさらに強化した「SX-WINDOWver.3.0」を標準装備。新たに、65,536色の自然色グラフィック表示を可能とした『グラフィックウィンドウ』※を搭載。またアニメーション動画をウィンドウ上で表現でき、手軽にコンピュータアニメーションが楽しめる『CGAウィンドウ』、さらに従来のエディタのイメージを一新、高度な日本語文書作成をサポートするSX-WINDOW対応の高機能日本語マルチフォントエディタを標準装備。アウトラインフォントの展開もさらに高速化が図られています。

※SX-WINDOW上の512×512ドットのエリア内で表示可能。

GUIに対応する大容量メインメモリを搭載。

メインメモリは標準で4Mバイト、複数のアプリケーションをウィンドウ上で同時に使用するなど大量のデータ処理に対応。また本体内の増設で、I/Oスロットを使用せず最大12Mバイトまで拡張できます。拡張したメモリはすべて32ビットバスによる高速アクセスが可能、優れた拡張環境でシステムパワーアップをサポートします。

※メモリ増設には、4MB内部増設RAMボードCZ-5BE4標準価格54,800円(税別)、4MB増設RAMモジュールCZ-5ME4標準価格49,800円(税別)をご使用ください。なおCZ-5ME4はCZ-5BE4上に装着します。

X68000シリーズの高機能を継承した上で、さらに使いやすさの向上を図ったコンパチビリティ重視設計※1、すぐに使える高機能ソフトを標準装備。

●25MHzでは速すぎるアプリケーションも、従来のクロック周波数(10MHz/16MHz)で動作可能なソフトコンパチ重視設計●65,536色同時発色の自然色グラフィックス(最大表示エリア512×512ドット)、1024×1024ドットの実画面エリアを持つ高解像度表示能力(最大表示エリア768×512ドット・カラー液晶ディスプレイ使用時※2は640×480ドット)、疑似高解像度スーパーインポーズ(インターレース方式/512×480ドット・専用ディスプレイテレビ使用時)を装備した高精細度自然色グラフィックス機能。●外部MIDI音源もコントロール可能※3、ウィンドウ上で手軽にコンピュータミュージックが楽しめるMIDI音源対応デバイスドライバ搭載●ステレオ8オクターブ8重和音FM音源、ADPCM搭載●プリンタ、RS-232C、SCSI、オーディオ入出力、イメージ入力など多彩なインターフェイスを装備。●日本語変換効率や操作性を高めた日本語フロントプロセッサASK68Kver3.0搭載。●従来のエディタのイメージを一新したSX-WINDOW対応の高速多機能日本語マルチフォントエディタ標準装備●日本語マルチフォントエディタ中に貼り付ける絵やグラフなどが簡単に作成できるグラフィックパターンエディタ●MIDI対応のX-BASIC。

※1 アプリケーションソフトおよび周辺機器のうち、一部動作しないものがあります。詳しくはシャープお客様相談窓口にお問い合わせください。

※2 10.4型カラー液晶ディスプレイLC-10C1-H標準価格598,000円(税別)、接続ケーブルAN-1515X標準価格4,200円(税別)をご使用ください(SX-WINDOW対応アプリケーションのみ。色数に制限があります)。

※3 別売のMIDIインターフェイスが必要です。

130mmFD(5.25型)マンハッタンシェイプシリーズ

■X68000伝統のマンハッタンシェイプを継承 ■130㎜FDD(5.25型)2基搭載 ■80MBハードディスク内蔵(CZ-510C)※ ■マウス・トラックボール標準装備 ■ASCII準拠フルキーボード採用

※CZ-500Cには、80MB内蔵用ハードディスクドライブCZ-5H08/160MB内蔵用ハードディスクドライブCZ-5H16を用意しています。

X68030
32bit PERSONAL WORKSTATION

本体+キーボード+マウス・トラックボール
130㎜(5.25型)FDタイプ
CZ-500C-B(チタンブラック)標準価格398,000円(税別)
HD内蔵 CZ-510C-B(チタンブラック)標準価格488,000円(税別)
14型カラーディスプレイ
CZ-608D-B(チタンブラック)標準価格94,800円(税別・チルトスタンド同梱)

90mmFD(3.5型)コンパクトシリーズ

■32ビットのハイパワーを凝縮したコンパクトフォルム ■2DD対応90㎜FDD(3.5型)2基搭載 ■80MBハードディスク内蔵(CZ-310C)※ ■マウス標準装備 ■コンパクトキーボード採用

※CZ-300Cには、80MB内蔵用ハードディスクドライブCZ-5H08/160MB内蔵用ハードディスクドライブCZ-5H16を用意しています。

Compact

本体+キーボード+マウス
90㎜(3.5型)FDタイプ
CZ-300C-B(チタンブラック)標準価格388,000円(税別)
HD内蔵 CZ-310C-B(チタンブラック)標準価格478,000円(税別)
14型カラーディスプレイ
CZ-608D-B(チタンブラック)標準価格94,800円(税別・チルトスタンド同梱)

X68030
32bit PERSONAL WORKSTATION
&
X68000
PERSONAL WORKSTATION・XVI

X68030/X68000を手に入れたら、やっぱり他のユーザーがどんな風に使っているのか気になるもの。ということでEXEクラブは、そんなあなたのための、他の68ユーザーとのコミュニケーションをバックアップする、情報交換の場です。

本体同梱の入会申込ハガキを送るだけで、自動的に無料入会。さらに下記の特典付き。

メリット 1 会員ナンバー入りオリジナル会員電卓がもらえる。

メリット 2 各種フェアご優待・イベント案内等、数々の特典がある。

●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。

■お問い合わせは… シャープ株式会社 コンシューマーセンター 西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

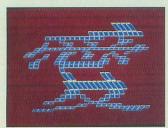
資料請求券 X68030 Oh!X 1係

C O N T

〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／山田純二　豊浦史子　高橋恒行　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　吉田賢司　朝倉祐二　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　司馬　護　清瀬栄介　石上達也　柴田　淳　瀧　康史　横内威至　進藤慶到　●カメラ／杉山和美　●イラスト／山田晴久　江口響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら

特集　Z-MUSICシステムver.2.0

ストリートファイターIIダッシュ

餓狼伝説 2

"実戦！" ゲーム作りのKNOW HOW

DōGA CG アニメーション講座

NUCLEAR WAR

表紙絵：塚田　哲也

1994 JAN.
1

E　N　T　S

■広告目次

先が、面白くなる。

ウィンドウ環境のプラットホームを確立、SX-WINDOW ver.3.0

■実画面：1,024×1,024ドット、表示画面：768×512ドット

●この画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコン等は、SX-WINDOW ver.3.0がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中のエディタで表示している文字のフォントはZeit社の、『書体倶楽部』のフォントを使用しています。

シャープ株式会社
●お問い合わせは…コンシューマーセンター西日本相談室　〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部システム機器営業部　〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

SHARP

目の付けどころが、シャープでしょ。

…に見たGUIの新展開。

発展性のあるプラットホームとしてのウィンドウシステム、
SX-WINDOW ver.3.0が提供する新たなGUI環境が
さらなるウィンドウ時代を予見する――――――。
国産オリジナルウィンドウとしての意味、未来への確かなビジョン、
ユーザーインターフェイスや高速化へのゆるぎない探求が
ここに凝縮されています。
65,536色表示はもちろん、さまざまな画像フォーマット対応、
イメージデータのコピー&ペースト、
動画、音楽/音声再生をサポートするマルチメディア環境。
そして、何よりもこれらが密接に連携して
統合的にハンドリングできるエキサイティングな環境を創造しています。
未来を照準に入れたウィンドウアーキテクチャ、
そのインテリジェンスがいよいよX68030/X68000シリーズで享受できます。

❶マルチフォントエディタ編集例。文字ごとに文字種、文字の大きさの指定、修飾が可能で、イメージデータの貼り付けもOK。

❷CONFIG.SYSやAUTOEXEC.BATなどの編集に便利な「エディタ」モードの例。このように日本語マルチフォントエディタは、用途に合わせてカスタマイズできます。

❸❶の画面をプリンタで印字した例。対応プリンタも増えました。(カラー印刷は誤差分散により65,536色対応)

❹「パターンエディタ」で作成したデータを、背景に設定できます。

❺バージョンアップした日本語フロントプロセッサASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❻アイコンデータや背景データを作成する「パターンエディタ」。文字の貼り付けなど、編集機能も一段とフレンドリーに。

❼オリジナルに作成したアイコンパターンの例。

❽512×512ドットの範囲内で65,536色の表示が可能。

❾さまざまなグラフィックフォーマットに対応しています。

❿任意のサイズに縮小・拡大表示可能。

⓫異なる画像フォーマットへのコンバートができます。

⓬「CGAウィンドウ」、65,536色(最大)のコンピュータアニメーション表示が可能です。

X68030

X68030 Compact

X68000 XVI

X68000 XVI Compact

SHARP

目の付けどころが、シャープでしょ。

For X68030/X68000series APPLICATION SOFTWARE

◎定評のあるGUI対応のウィンドウワープロ。

EGWord SX-68K

CZ-271BWD　1月発売予定

ウィンドウワープロとして評価の高いEGWordのSX-WINDOW対応版。
キャラクタベースのワープロを超えたグラフィカル・ユーザーインターフェイス(GUI)による手軽なDTPソフトとしても優れた表現力を発揮します。定評ある日本語入力方式(EGConvert)によるインライン入力、文書互換を実現するEDF形式もサポートしています。

NEW

●禁則処理を生かした美しく、読みやすい文書作成：文字間隔を自然に美しく配置。さらに均等禁則など、豊富な禁則処理によるきめ細かな調整が可能。

●作図感覚で罫線・作表作業も快適：マウスによる作図感覚の作表、また豊富な線種で、行や列を気にせずに文字と文字の間に罫線が引けます。表組も自在に編集できるとともに、罫線で囲まれたブロック単位で網掛けも可能。

●DTPに迫る多段組、レイアウト表示：段組は2段から5段まで設定でき、段間隔の調整や段組線の表示はもちろん、自由な位置での改段も可能。レイアウト表示もOK。

●様々なグラフィックデータやテキストデータの貼り込みが可能：他のソフトで作成された色々な画像データ(GScript形式)をEGWordの文書に取り込めます。もちろん取り込んだ画像の編集もOK。

●短文・書式登録でルーチンワークの負担を軽減●充実した国語辞書機能に優れた専門辞書をプラス●実用性の高い逐次自動変換方式を採用●ウィンドウの特性を生かした優れたユーザーインターフェイス●ルーラによるスピーディ&イージーな書式設定●文書互換を実現するEDF(Extended Document Format)形式をサポート。

*5MB以上の空きのあるハードディスクが必要です。(4MB、ver.2.0)

◎待望のSX-WINDOW開発支援ツール。

SX-WINDOW 開発キット Workroom SX-68K

CZ-288LWD　1月発売予定

SX-WINDOW用のソフト開発に必要な開発ツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解ができる33種のサンプルプログラム付き。また各マネージャ解説と関数リファレンスの詳細なマニュアルも装備しています。

NEW

※メインメモリ4MB以上、SX-WINDOW ver.2.0以上、C compiler PRO-68K ver.2.1が必要です。

〈キット構成〉

■開発ツール

●SXデバッガ：SX-WINDOW上で複数のプログラムを同時にデバッグすることができるソースコードデバッガ。

●リソースエディタ：SX-WINDOW上のリソースをリソースタイプごとの編集ウィンドウでビジュアルに作成・編集が可能。

●リソースリンカ：Cコンパイラやアセンブラで作成したリソースデータファイル(オブジェクトファイル)をリンクしてリソースファイルを作成。

●サンプルメイク：サンプルプログラムのコンパイル作業をSX-WINDOW上から、XCver2.1のMAKE.Xを呼び出して、自動実行する簡易メイクユーティリティ。

■サンプルプログラム

●基礎編(23種)：各マネージャの基本的な機能のみを用いた基本動作の理解。

●応用編(4種)：基礎編での基本機能を応用した簡単なアプリケーションの作成。

●実用編(6種)：基礎/応用編での機能を駆使した、実用的なアプリケーションの作成。

■その他ファイル

●インクルードファイル：Cコンパイラとアセンブラ用の関数定義、データ定義ファイル。

●ライブラリファイル：Cコンパイラ用関数ライブラリ。

[マニュアル]

●ユーザーズマニュアル●プログラマーズマニュアル●SXライブラリマニュアル

X68030
32bit PERSONAL WORKSTATION

その先のシーンへ。

●65,536色対応、動画ウィンドウ標準装備。

SX-WINDOW ver3.0 システムキット

CZ-294SS(130㎜FD)/CZ-294SSC(90㎜FD)各標準価格19,800円(税別)

自然描画に迫る美しい表現が可能な65,536色表示のグラフィックウィンドウを装備。さらにグラフィックウィンドウ内でのアニメーション動画表示、各種グラフィックデータのコンバートも実現しました。またイメージデータの貼り付けなどをサポートした日本語マルチフォントエディタを始め、クリエイティブワークを支援する数々の便利機能を装備、Human68k ver.3.0システムディスクも付属しています。

※メインメモリ4MB以上必要です。SX-WINDOW ver.1.0/1.1/2.0をお持ちの方には有償バージョンアップを行っています。

●「SX-WINDOW開発キット」のサポートツール。

開発キット用ツール集

CZ-289TWD　2月発売予定　NEW

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。(2MB、ver.2.0)

●SX-WINDOW対応ドローイングツール。

Easydraw SX-68K

CZ-264GWD　標準価格19,800円(税別)

ホビーからビジネスまで幅広い分野で活用できる、待望のドローイングツールです。イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは、他のSX-WINDOW対応アプリケーションでも利用でき、企画書やプレゼンテーション資料の作成をサポートします。またレーザープリンタドライバを付属、このドライバはSX-WINDOW対応の他のアプリケーションでも利用することができます。(4MB、ver.3.0)

●SX-WINDOWを楽しく使うためのアクセサリ集。

SX-WINDOW デスクアクセサリ集

CZ-290TWD　標準価格14,800円(税別)

SX-WINDOWをさらに便利に、楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、アドレス帳、電子手帳通信ツール、パズルなど12種類の豊富なアクセサリが収められています。

1キーノート2スクリーンセーバ3スクラップブック4ミュージックボックス5ハイパーリンク(電子手帳通信ツール)6アドレス7スケジューラ8ウィンドウアイコニファイ9ソフトウェアキーボード10パズル11ファイルサーチ(ファイル検索ツール)12フォントリンカ。(2MB、ver.3.0)

●マルチタスク機能をはじめ、通信環境がさらに充実。

Communication SX-68K

CZ-272CWD　標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。(2MB、ver.1.1)

●ウィンドウ対応グラフィックツール。

Easypaint SX-68K

CZ-263GWD　標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。(2MB、ver.1.1)

●FM音源サウンドエディタ。

SOUND SX-68K

CZ-275MWD　標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。(2MB、ver.1.1)

●SX-WINDOW対応になってさらにパワーアップ。

倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293AW(130㎜FD)/CZ-293AWC(90㎜FD)各標準価格6,800円(税別)

倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。(2MB、ver.1.1)

●X68030/X68000対応　NEW

CZ-295LSD　標準価格44,800円(税別)

C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フロートライブラリを付属しています。

※メインメモリ2MB以上が必要です。
※C compiler PRO-68K/ver.2.0/ver.2.1をお持ちの方には有償グレードアップサービスを行います。

※(2MB,ver.1.1)の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。
※発売予定のソフトの画面は実物とは異なる場合があります。

●お問い合わせは…シャープ㈱電子機器事業本部(液映)システム機器推進プロジェクトチーム　〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ　シャープ株式会社

SHARP

X68ゲームコンテスト開催。腕に自慢のゲームフリークたちへ。挑戦を待つ!

サバイバル

開催期間:1993年12月〜1994年1月

(全国で続々開催.!)

TOWNS TAKEBE
……………………12月19日(日)

(株)ウェーブ・アイ 湘南台店
……………………12月23日(祝)

(株)栄電社 テクノ名古屋店
……………………12月23日(祝)

(株)ニノミヤ ニノックス・コア
日本橋店…………12月23日(祝)

ダイイチCOM CITY
……………………12月26日(日)

(株)満開製作所 於:シャープ市ヶ谷
ビル エルムホール…12月26日(日)

SRGスタンバイ Ver.3
……………………1月8日(土)

ラオックス株式会社
ザ・コンピュータゲーム館‥1月22日(土)・23日(日)

同時イベントとして●ストリートファイターIIキャラクタイラストコンテスト●ストリートファイターIIコスチュームプレイコンテストなど、おもしろイベントも開催予定。

■詳細は、お近くのX68販売店または下記へお問い合わせください。

■主催:シャープ株式会社電子機器事業本部システム機器営業部
電話(06)621-1221(大代表)

■協賛:CAPCOM® 株式会社カプコン
株式会社ボークス(「1/8春麗人形」製造元)

Magical
COMPANY
今再び、伝説の男達が帰ってきた・・・
●商品サンプル
これが魔法オリジナルジョイパッドだ！
X68000の人気格闘
「餓狼伝説」が帰ってくる。
ネオジオ版にもなかった
闇の帝王と三闘士の操作を可能にし、
さらに、なんと専用の
4つボタンジョイパッドをソフトに同梱。
あらゆる面で超パワーアップを果たした
魅力あふれる12大ファイター達が
"新たなる闘い"へと挑む。
最強の対戦格闘「餓狼伝説2」
いよいよ年末発売。
NEO GEO
餓狼伝説2
がろうでんせつ2 —— 新たなる闘い ——
©1993 魔法株式会社 / © 1992 SNK
X68000シリーズ
12月23日発売予定
標準価格9,800円（税別）
●SHARP X68000 / 30シリーズ及びXVI専用●5" 2HD（3.5インチモデルには対応しておりません。）●要2MB●ハードディスクインストール可能●内蔵FM音源対応●純正MIDI音源対応（SOUND CANVAS SC-55、MT-32 いずれもローランド社製）●キーボード及びジョイスティック※3操作対応
ALL THAT これでもか！
Point 1 なっ、なあんと！初回生産分に限り、餓狼伝説2専用魔法オリジナル4つボタンジョイパッド※1をソフトに同梱。はっきり言ってお得です。
Point 2 ネオジオ版でも使えなかった闇の帝王と三闘士操作が可能！※2。操作方法は餓狼伝説SPECIALとほぼ同じだよ。
Point 3 2P対戦、途中参加、同キャラ対戦可能はあたりまえ。加えてさらにVSモードを追加！対戦プレイで熱く燃えろ！
Point 4 MIDIサウンドも大幅にパワーアップ（当社比2.5倍）！超ド級のリアルサウンドがサク裂する!
※1：初回生産分に限りソフト本体に同梱（1個）。なお、ジョイパッドの故障、不都合などに対するお問い合わせには、一切お答えできません。また、ジョイパッドのみの別売はいたしませんのでご了承ください ※2：闇の帝王と三闘士の操作はVSモード内に限ります。また、闇の帝王と三闘士の超必殺技は使用できません。 ※3：2つボタン、3つボタン、4つボタン（本体付属専用ジョイパッド）及び、電波新聞社製メガドライブコントローラー変換アダプターにより6つボタンジョイスティックに対応しています。
企画・開発・発売 魔法株式会社 〒651 神戸市中央区葺合町馬止1-10 マジカルビル PHONE.078-261-2790 FAX.078-261-2792
販売 ビクターエンタテインメント株式会社

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

最終回 さひたふさん ID:JH261272

最終回を飾るのは、プログラム開発がお好きだという、さひたふさん。ワークステーションなどで有名なUNIX系の各種ツールなど、そのソフトの好みも、実に「通」。X68000をスマートに使いこなす通信生活などなど、いろいろ教えていただきましょう。

=基本データ=
■使用機種：本体　X68000 PRO－HD（CZ－662C－GY）
　　　　　：ディスプレイ　CZ－602D－GY
■周辺機器：モデム　11400bpsモデム
　　　　　：プリンタ　BJ－10v
■使用開始時期：1989年5月中旬頃
■アクセス頻度：3日に1度
■おすすめX68000用フリーソフト：μEmacsエディタ
■定期巡回ボード：（SIG）SHARP－HOTLINEの「Oh／X68K」

■X68000を選んだ理由は？

1．初めて触ったパソコンがMZ－80Bで、それ以降SHARPのパソコンしか目に入ってこなかった。
2．X68000シリーズが好きなのに理由はいらいない／／／そこにX68000があるから。(^^;

■おもにどんな用途に使われていますか？

プログラム開発、環境を整えるためのプログラムばかり作っています。こんなのがあれば、あとあと便利だろうというものばかり作っていて、なにか特定の目的のためのプログラムをしている訳ではありません。(^^;
あとはパソコン通信。最近はほとんどこれです。

■好きなX68000用フリーソフトウェアは？

μEmacsエディタです。マイクロじゃないフルセットの GNU Emacsを使いたいのですが、ハードディスクの容量が足りないので…。

■HOTLINEに入会したきっかけは？

友人の勧めで。京都にアクセスポイントがあり、入会もお店でスタータキットを買ってきてすぐに使えたから。

■ニックネームの由来は？

本名から。(^^;

■HOTLINEを何に活用されていますか？

新しい周辺機器の性能やショップ価格などの情報収集など。とくにXユーザーは周りに少ないので、パソコン通信などでの情報は欠かせないものになりました。

■X68000、こんな使い方が面白い！

TeX組版ソフトは面白いですね。ワープロを使ってる人には使い難いものかもしれませんが、慣れるとこちらの方が使いやすいと思います。使い慣れたエディタを使えるという利点もあります。
その他、fishとか。command. xのかわりに使うコマンドインタプリンタで、UNIXのCシェルライクな操作ができます。
SHARP－HOTLINEやCZ－CLUBで行っているディスク回覧に参加すると入手できます。
CZ－CLUBでは、MOディスクを回覧しようという話も出てきましたし、これでTeXやNemacsなど巨大なフリーウェアの回覧もやりやすくなると思います。皆さんも参加しませんか？

■X68000ユーザーに知らせたいHOTLINEのコーナーは？

やはり SHARP－HOTLINE でしょう。

■あなたにとって、J&P HOTLINEは？

ブラックコーヒー。寝る前の私の睡眠時間を奪っていくから。(^^;
（寝る前にアクセスしています）

J&P HOT LINEへの
ご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000＋¥90(消費税3%)＝¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは――
〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOTLINE事務局宛　TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

								岸和田店	☎(0724)37-1021	和歌山店	☎(0734)28-1441
渋谷店	☎(03)3496-4141	津田沼店	☎(0474)72-5211	寺地店	☎(0762)47-2524	高槻店	☎(0726)85-1212	さんのみや1ばん館	☎(078)231-2111	和歌山南店	☎(0734)25-1414
町田店	☎(0427)23-1313	越谷店	☎(0489)66-1221	大須店	☎(052)262-1141	くずは店	☎(0720)56-8181	西宮店	☎(0798)71-1171	学園前店	☎(0742)49-1411
八王子店	☎(0426)26-4141	伊勢崎店	☎(0270)21-1121	テクノランド	☎(06) 634-1211	千里中央店	☎(06) 834-4141	伊丹店	☎(0727)77-5101	奈良1ばん館	☎(0742)27-1111
立川店	☎(0425)36-4141	焼津インター店	☎(054)626-3311	メディアランド	☎(06) 634-1511	摂津富田店	☎(0726)93-7521	姫路店	☎(0792)22-1221	新大宮店	☎(0742)35-2611
三鷹店	☎(0422)31-6251	にいがた1ばん館	☎(025)241-3711	コスモランド	☎(06) 634-3111	寝屋川店	☎(0720)34-1166	京都寺町店	☎(075)341-4411	郡山インター店	☎(07435)9-2221
相模原店	☎(0427)55-2831	富山店	☎(0764)22-5033	U.S.LAND	☎(06) 634-1411	枚方バイパス店	☎(0720)48-1211	京都近鉄店	☎(075)341-5769	田原本店	☎(07443)3-4041
本厚木店	☎(0462)25-5151	金沢店	☎(0762)91-1130	ビジネスランド	☎(06) 348-1881	藤井寺店	☎(0729)38-2111	大久保バイパス店	☎(0774)44-1211	熊本店	☎(096)359-7800

Z-MUSICシステム Ver.2.0

ついにMUSICシステムの正式バージョンアップ版が登場します。
X68000の音源ドライバとしてさらに使いやすく高機能なものになりました。

ver.1.0/1.1からのバージョンアップ内容

PCM8対応AD PCM同時発音8声音量可変
モジュレーション用波形メモリ搭載
PCMバンクに対応
ステップエディット系コマンド追加
X68030完全対応ユニバーサルバージョン
RS-232C対応版収録
POLYPHON対応版収録
再生専用機能縮小版収録
Cコンパイラ用ライブラリ完成
AD PCM加工機能強化
さらにクオリティを高めたAD PCMデータ
もちろん、全ソースプログラム付属&ライセンスフリー

SOFT BANK ソフトバンク株式会社／出版事業部
〒103 東京都中央区日本橋浜町3-42-3 TEL03-5642-8100

[第32回]
お年玉

江口 響子

響子inCGわ〜るど

あけまして，おめでとうございます。

おかげさまで，この連載も3回目のお正月を迎えました。お会いしたことはないけれど，いつも読んでくださる皆さんには，感謝の気持ちでいっぱいです。ときどきいただくお便りは，とても励みになります。小さな部屋の片隅に置いてあるパソコンのX68000で，CGを作り続けていて本当によかったと思います。

さて，皆さんは，お年玉をもらいましたか? それとももう大きくなって，お年玉をあげるほうにまわっているのでしょうか。

初めて私がお年玉をもらったのは，ものごころのつき始めた3つか4つの頃だったそうです。あまりに幼かったので，お金がどういうものなのか理解できなかったのですが，袋にかわいいイラストが描いてあり，嬉しかったのを覚えています。母は郵便局に私名義の口座を作り，そこに最初のお年玉を貯金しました。以来，小学生，中学生，高校生と，お年玉のほとんどを貯金するという習慣が続きました。

大学に入ってから，貯めたお年玉を使う機会がやってきました。富田勲さんやYMOに憧れて，どうしてもシンセサイザが欲しくなったのです。お年玉とその頃していた家庭教師のアルバイトで貯めたお金とを合わせて，ローランドのSH-2というシンセサイザを買いました。フロッピーディスクドライブやメモリなどの記憶装置がないアナログシンセサイザで，しばらくのあいだ，思いついた音を作ったりして遊んでいました。が，使いこなせないうちに，同級生に譲ってしまいました。

短い間ですが，そのことがマシンで何かものを作るというきっかけになったように思います。いまのCG制作へとつながっていったのです。

数年間は，趣味でCGを作っていました。もちろんパソコンで。そのうちに，イラストとしてCGを仕事にするようになり，最近，テレビでCGの制作をする機会に恵まれました。ところが，パソコンだけを使っていたときと，ぜんぜん雰囲気が違います。動くお金が1桁も2桁も違う世界。ものを作る才能に加え，お金を集める才覚も必要です。たくさんのお金を集め，より高性能のコンピュータソフトを手に入れることがすなわち，よいCG作品を作ることのように見えました。CGの質がコンピュータの性能に依存している以上，しかたないのかもしれません。しかし，もの作りとお金集めが混沌としているそうした世界は，パソコンで細々とCGを作っていた私にとって，ちょっと馴染みにくいものでした。

規模は小さいかもしれないけれど，パソコンには，じっくりと自分の好きなCGを作ることので

きる環境があります。もちろん，CGだけではありません。自分が本当に作りたいと思った曲を作る，誰もが使える小さなツールをプログラミングする，気のおけない友人とゲームを作る……そんな環境にじっくりと自分を浸す，満ち足りた時間があります。

とうにお年玉をあげる立場になってしまった私にとって，パソコンによって得られるこうした時間は，お金で買うことのできない大切なもののように思えます。

パソコンで，X68000で何か作ってみたいと考えている読者の皆さんにとって，1994年が素敵な年になりますようにお祈り申し上げます。

今回のCGデータ

1920×1536ピクセル
1670万色フルカラーを4×5ポジに出力
総物体数　147　光源　1　折り鶴はポリゴン
使用ソフトは，サイクロン
ポリゴンデータ作成には，Z's TRIPHONY DIGITAL CLAFT
マッピングデータ作成には，MATIER

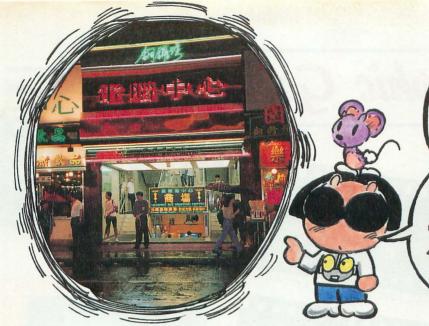
another
CG world
in Hong Kong
行きも 帰りも
B747－400
テクノ ジャンボ
成田→香港は
約3時間半
ANA
Economy Class
EGUCHI/KMS
NH910
25-SEP-93
NRT
1410
48H
362
ANA
All Nippon Airways
BOARDING PASS
搭乗券
92年11月
から新しくなった
パスポート
日本国
旅券
JAPAN
PASSPORT
中華人民共和国
九龍
にぎやか
なのは
この
あたり
ランタオ島
香港島
4年後の
1997年7月1日
に 英国から
中国に 主権
が 返還されるのか…
香港は、九龍半島と香港島、
ランタオ島などの島で成り立っている。
香港には、
日本の日銀にあたる
中央銀行がありまへん。
香港上海豊銀行
と香港渣打
銀行が、額面
は同じでも、デザインの違う紙幣を
発行しているのだ。
Standard Chartered Bank
香港渣打銀行
One Hundred
Dollars
ONE
HUNDRED
DOLLARS
TEN
DOLLARS
Ten Dollars
$100
$100

1HK$
(≒¥15)
でOK
なのよ

世界中で
ここ香港にしかない
2階建て路面電車
のトラム。乗車口には
「上」。降車口には
「落」と書いてある。
なるほど

MTR (Mass Transit Railway)
地下鉄のキップは、JRのイオ・カード
みたい！

自動改札口で回収され、
何回もリサイクルされるのよん。

地下鉄の通路で見つけた
SHARPの広告。なんか
あやしげなカラオケセットの
ようだが…

九龍と香港島をむすぶのは、フェリー。

1等船室（といっても1.5HK$ ≒¥22）
の中…

PEARL CINEMA
明珠戲院
JADE CINEMA
翡翠戲院
國際咖喱館
INTERNATIONAL
これは 映画館 のネオン
東方院線
明珠戲院
PEARL THEATRE
嚴禁吸煙 Smoking Prohibited
9:30 p.m.
Admit one only
Adm. $40
№ 03360
C
23 SEP 1993
全席座席指定
香港映画「陰陽法王」を観た。「チャイニーズ・ゴーストストーリー」で有名な、ユーレイ美女、王祖賢（ジョイ・ウォン）主演。
字幕は、英語と中国語。
中国語は、漢字は同じだが発音のちがう方言がいくつかあって香港では、広東語が日常語。
いろいろな ネオンの看板がこれでもか これでもかと道にはり出している。
歯医者のネオンなんかもあったりする。
しゃべるのは広東語
ほとんどの建築物の足場は、「竹」で組んである。
ぐにゃぐにゃしてて こわいなー ビルは、まっすぐなんだけどね…
もちろん 飲茶もいっぱい食べた。
文字だけでは、中身がよくわからないので フタをとって もらってえらぶ。
叉焼飽
粉果
山竹牛肉

これは、電脳中心(センター)。
香港には、日本のアキハバラみたいな電気街っつーのはないかわりに…ラジオデパートのようにいろいろなお店が入っている電脳中心があちこちにある。
これは、銅鑼湾(トンローワン)にあるもの

中には、
ボード&ソフトを売る店
IBM-PCなど
マレーシアやシンガポール製のもの

Mac専門店

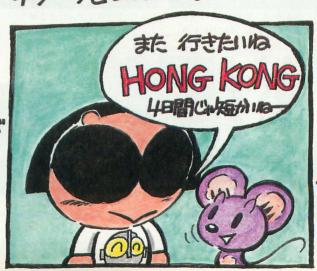
AMIGAやMACの本屋
HOW TO USE MAC

GAME屋
日本のGAMEがいち早く店頭に並ぶ
「餓狼伝説2」のポスターも貼ってあった。

なぜか花屋やボディコンの洋服屋もある。

それと、マンガ店
高橋留美子的漫畫
高橋留美子さんや鳥山明さんは超人気。中国語に訳されてずらーっと並んでいる。

しかし、電気製品といえば…
街中で、ケータイ電話をかけている人が多い。なんでも基本的な通話料はタダなんだそーだ。

通信したい人にはうらやましいかぎり…
ごくふつうのOLの女の子がもっていたりする…

KYOKO '93 10-23

EPA2の使い方の2回目です。表現のやり方次第で，得られる効果は大きく違ってきます。

写真A　だんだん遠ざかっていく宇宙戦闘機。今回はこのカットのなかにビーム光を入れてみる

(左から)写真H，I，J，K　ビームが貫通して開いた穴から放電現象が始まり，稲妻が拡がっていく

写真B　4フレーム目の画像に1発目のビーム光を描き込む。この色のRGB値は赤8，緑25，青12である

●手前と奥とでは太さが違ってくるので，奥行きのある絵の場合は3次元的なパースを計算する必要がある

写真N-1　EPA2で光らせる前

写真N-2　EPA2で光らせたあと

写真C　「Filter」の「Snow Fire」で光らせる

写真P-1　EPA2による円

●よく見ると，違いがわかる

写真P-2　ポリゴンによる球

写真S-1　光源を暗くする

●光源の明るさにより印象が異なる

写真S-2　光源を明るくする

写真Q　この飛び散る火花もポリゴンで作成している

写真T　暗いカットを明るいカットのあいだに挿入すると，連続して見たときに画面全体を点滅した感じになり，明るさがより強調される

SOFTWARE INFORMATION

格闘ゲームの並んだ先月号から一転，今月初登場はパズルゲームとシミュレーションです。激しい闘いのあとの疲れた身体にはひと口の砂糖菓子のようにおいしいかな？いやいやあまくみると，泣きをみるかも。

キーパー

世界の奥の誰も知らない静かな村に，突然起こった大事件。たったひとつの泉のなかに海の生物の化石が埋め込まれた石板が降ってきた。さあ，石板を消して泉を守れ！

ルールは簡単。この不思議な石板は，同じ色か同じ化石のものが3つ以上並ぶと消えてしまうのだ。押すことも引くこともできるので，どんどん並べ替えていけばよい。

キャラクターの「プクル」と「ピクル」の動きが可愛いくて，キミの彼女も気に入りそう。

ふたりで遊ぶときには「協力モード」と「対戦モード」が選べるので，一緒にべたべた仲よくプレイするか，アツく競うか，遊び方も2通りある。

「LOGIN」誌上で行われているソフトウェアコンテストのグランプリ受賞作をもとに，サクセスが開発している。発売は12月25日。

X68000用　5″2HD版　8,800円(税別)
電波新聞社　☎03(3445)6111

闘いの季節到来？

		(前回順位)
1.	ストリートファイターⅡダッシュ	1
2.	餓狼伝説2	4
3.	ドラゴンバスター	—
4.	スタークルーザーⅡ	5
5.	SX-WINDOW開発キット	7
6.	ぷよぷよ	—
	EG Word	8
8.	卒業～GRADUATION～	10
9.	マージャンクエスト	—
10.	SimCity2000	—

12月号のハガキから「期待している新作ソフト」の集計です。

先月に続いて1位の「ストリートファイターⅡダッシュ」。実はこれを書いているいまは，すでに発売されていますので，最近になって編集部に届くハガキでは，「買って満足のソフト」のほうで名前が挙がっています。

さて，こちらの集計ではポイント数は特に掲載していませんが，今回は1位と2位はほとんど同じポイント数で3位以下を大きく引き離しました。しかもジャンルも同じ格闘もの。1993年の年末は，格ゲーファンにとっては幸せな1年の締めくくりになったのではないでしょうか。ゲームの出来などについては今月号のレビューでも紹介していますが，この2つのビッグタイトルは来月号でもひき続き，さらに掘り下げて紹介する予定です。読者のみなさんのご感想やご意見もお待ちしています。

3位の「ドラゴンバスター」はビデオゲームアンソロジーシリーズの最新作。このシリーズはコンスタントに人気を獲得しています。「ドラゴンバスター」についても，今月号のレビューを読んでくださいね。

先月号で発売予定をお知らせした「ぷよぷよ」もさっそく6位と，期待を集めています。メガドライブで大ヒットした落ちモノゲームですが，編集部でも一時期対戦ブームになりました。X68000版の詳細は，残念ながらまだ発表されていません。

格闘ゲームが続いたあとは，ちょっと違うジャンルの新作の発売が続きそうです。闘いの爽快感に身を委ねたあとは，じっくり頭を使ってみるのもいいかもしれませんね。

卒業〜GRADUATION

「卒業」は，PC-9801版が大ヒットした「教育」シミュレーションゲーム。女子高の新任教師となったアナタは，可愛い教え子たちを無事に卒業させなければならない。

登場するのは5人の個性的な女の子。タイプがまったく違うだけに，それぞれの性格に合わせた指導をしてやらないと困ったことになる。オトナでもなく子供でもない魅力的で微妙なお年頃なだけに，彼女たちのまわりには危険がいっぱい。勉強嫌いの子や不良にあこがれる子，病気になっちゃったり，失恋して学校サボっちゃったり……。教師は心配が絶えないのだ。

そんな彼女たちを正しく導くためには，いろいろなことに気を配ることが必要。授業をサボらないように，うまくスケジュールを組む。品位を上げるために，しつけをする。生活指導や補習，面談や見回りもしなくては……。

学園生活には，創立記念祭や中間・期末試験，クラスマッチ，そして夏休みもある。それらを経験して彼女たちはどう成長していくのだろう。

X68000版では，画面デザインが多少変更されるらしい。発売は1月の予定。

X68000用　5″2HD版　価格未定
ブラザー工業(TAKERU)　☎052(824)2493

R.C.ロボット集+α

今月のハガキ集計でも「買って満足したソフト」ベスト10にランクインした「ロボットコンストラクションR.C.」のデータ集が発売された。

過去に開催されたバトル大会(NIFTY-Serve上で行われた2回を含む)の結果や，それに参加したロボットのデータ(約120体)が収められているほか，「R.C.」を楽しむためのツールも用意されている。

「R.C.」では，対戦するためのフィールドは3種類で，それぞれひとつの地形しかなかったが，このツールを使えば，その3種類のフィールドでさまざまな地形が楽しめるようになる。

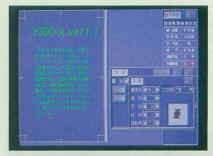
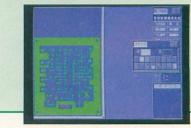
ロボットがうまく作れなかった人も，ここに収録されたロボットを研究すれば大丈夫だ。いままで「R.C.」を知らなかった君もセットで揃えよう。そして過去のチャンピオンを撃破して次のバトル大会を制覇するのだ。

X68000用　5″2HD版　1,800円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

Y300-A ver.1.1

1993年2月号で紹介した版下作成支援ソフト「Y300-A」。そのバージョンアップ版がこのほど発売された。今回は大幅な機能変更はないが，表示・作図の範囲指定や，図形の削除や移動(複写)，回転(回転複写)，拡大・縮小などの編集作業がマウスだけで行えるとか，イメージデータを直接プリンタ出力できるなど，細かい点での操作性が向上している。

X68000用　5″2HD版　34,800円(税別)
マグマソフト　☎0992(68)2286

TAKERU同人ソフト

ARTEMIS

連邦軍は実験戦闘艦ARTEMISを操り，帝国軍に立ち向かう。ポリゴンを使った，宇宙空間を舞台にした3Dシューティングゲームだ。

自機はシールドを装備していて，敵の攻撃にある程度(複数回)耐えられる。シールドがなくなり，さらにダメージを受けるとゲームオーバー。敵艦へは2門のレーザー砲で攻撃する。通常の宇宙空間ステージでは，自機の内部からと，自機後方からの2つの視点を選択できる。

X68000用　5″2HD版　1,800円(税込)

脳天伝説

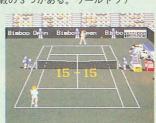
対戦タイプの麻雀ゲーム。もちろん勝てば相手の女の子が脱いでいく。最初のメニューで8人のなかから好きな子を選び，いざ勝負！

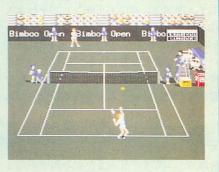
操作手段はマウスのみ。ツモと捨て牌は左クリックで行う。リーチやポン，チーなどをするときは，右クリックでウィンドウを開いてそこから選ぶ。とっても簡単で操作性はグッド。それにリーチをかけると，当たり牌を教えてくれて，初心者でも大丈夫だ。当たり牌を忘れてもウインドウを開けばいつでも教えてくれる。

X68000用　5″2HD版　2,500円(税込)

FIFTEEN ALL

クオータービュータイプのテニスゲーム。テレビの視点そのままだ。個性豊かなプレイヤー32人のなかから，好きなプレイヤーを選択できる。ゲームのモードは，ワールドツアー，エキシビジョン，対戦の3つがある。ワールドツアーでは，毎年12回のトーナメントをコンピュータと戦い，年間ランキングのトップを目指す。

作者のテニス歴は8年。これまでのテニスゲームに納得できずに，自分で作った思い入れたっぷりのゲームだという。

X68000用　5″2HD版　1,500円(税込)

ブラザー工業(TAKERU)　☎052(824)2493

発売中のソフト

★X Windows V11.5　マイクロウェアシステムズ
X68030用 3.5″2HD＋5″2HD版　30,000円(税別)

★Hyper Pixel Works　ブラザー工業(TAKERU)
X68000用　5″2HD版　19,800円(税込)

★ストリートファイターⅡダッシュ　カプコン　11/26
X68000用　5″2HD版　12,800円(税別)

★宝魔ハンターライム5　ブラザー工業(TAKERU)　12/10
X68000用　3.5/5″2HD版　1,500円(税込)

★R.C.ロボット集＋α Vol.1　ブラザー工業(TAKERU)　12/10
X68000用　5″2HD版　1,800円(税込)

新作情報

★Y300-A ver1.1　マグマソフト
X68000用　5″2HD版　34,800円(税別)

★餓狼伝説2　魔法株式会社　12/23
X68000用　5″2HD版　9,800円(税別)

★キーパー　電波新聞社　12/25
X68000用　5″2HD版　8,800円(税別)

★ドラゴンバスター　電波新聞社　12/中
X68000用　5″2HD版　5,300円(税別)

★宝魔ハンターライム6　ブラザー工業(TAKERU)　1/10
X68000用　3.5/5″2HD版　1,500円(税込)

★SX-WINDOW 開発キットWorkroom SX-68K　シャープ
X68000用　3.5/5″2HD版　価格未定

★SX-WINDOW 開発キット用サポートツール集　シャープ
X68000用　3.5/5″2HD版　価格未定

★卒業〜GRADUATION　ブラザー工業(TAKERU)　1/未
X68000用　5″2HD版　価格未定

★宝魔ハンターライム7　ブラザー工業(TAKERU)　2/10
X68000用　3.5/5″2HD版　1,500円(税込)

★麻雀航海記　ブラザー工業(TAKERU)　2/未
X68000用　5″2HD版　5,800円

★宝魔ハンターライム8　ブラザー工業(TAKERU)　3/10
X68000用　3.5/5″2HD版　1,500円(税込)

★マージャンクエスト　SPS
X68000用　5″2HD版　価格未定

★宝魔ハンターライム9　ブラザー工業(TAKERU)　4/10
X68000用　3.5/5″2HD版　1,500円(税込)

★ロボスポーツ　イマジニア
X68000用　5″2HD版　価格未定

★Traum　象スタジオ
X68000用　5″2HD版　価格未定

★鮫！ 鮫！ 鮫！　KANEKO
X68000用　5″2HD版　価格未定

★達人　KANEKO
X68000用　5″2HD版　価格未定

★エアバスター　KANEKO
X68000用　5″2HD版　価格未定

★サバッシュⅡ　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定

★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)

★スタークルーザーⅡ　アルシスソフトウェア
X68000用　5″2HD版　価格未定

★ぷよぷよ　SPS
X68000用　5″2HD版　価格未定

★GEOGRAPH SEAL　エグザクト　2/下
X68000用　5″2HD版　価格未定

お詫び　12月号で「餓狼伝説2」の発売日が12月22日とありましたが，12月23日の誤りでした。関係者の方々にご迷惑をおかけしたことをお詫びいたします。

熱き闘いの幕開け

期待の超大作,「ストリートファイターIIダッシュ」がついに発売された。現在の格闘ゲームブームを築き，コンシューマ機でも移植が好評なこの作品が，X68000にどのように移植されたか紹介していこう。

Shindou Noriyuki
進藤 慶到

もはや流行という言葉では片づけられない。「格闘ゲーム」という新ジャンルを創り上げ，いまなお現役かつ指標であり続けるゲーム史上最大級の名作,「ストリートファイターII」シリーズ(以下ストII)。

その波動はゲームセンターのみならず，家庭や学校，職場をも格闘場と化し，我々の平穏で健康的な日常ライフをものの見事に奪い取ってくれた。貴重なマネーも吸い尽くし，食費を削ってカルシウム不足，そして人間関係に復旧不可能なヒビを何本も作ってしまったという悲しい輩も存在する(らしい)。もちろん，いまなおストII貧乏という人種があとを断たない。「私はこれで学校を辞めました」「体売ってストIIの基板を買いました」という人も少なくはないだろう（ちょっとおおげさ)。

逆に，生涯のライバルというべき存在をストIIの対戦プレイから得たという熱い体験談や，ふらりと乱入した台の相手がたまたまタイプの女性だったりして，お互いに恋が芽生えてしまったという，うらやましい話もレポートされている。

そう，ストIIとはさまざまな人生模様を演出してきたゲームなのだ(強引)。

いよいよ登場，X68000版

ついにこのときがきてしまった。そう,「ストリートファイターIIダッシュ」がX68000版として発売されたのだ。あのストIIの波動が，我々のもっとも愛するパーソナルマシン上で完全に再現されている。もう待ってましたというほかに言葉が見当たらない。

ストIIには現在4つのタイプが存在する。X68000にはこのうちのストIIダッシュが移植されたわけだが，これはあえて正解だったと私はいいたい。もっとも遊びやすく，かつ丁度よいバランスが保たれているのが，このストIIダッシュだと思えるからだ。ターボは速すぎるのでついていけないとか，スーパーはベリーイージーランクでも難しくてクリアできない(実は私)，という人は多い。そんなプレイヤーも心配無用。

私の個人的な意見としては，ターボは色が好みじゃないし，速すぎてザンギエフの吸い込みが難しい(ヘタなだけだが)。スーパーは1人プレイでやってても面白くないし，だいいち爽快感がない。というわけでダッシュが移植の対象として選ばれたことに不満はない。

熱い男の肉弾戦。それがストIIだ！

哀れ，スクリューパイルドライバーの餌食

見よ，この超絶移植

忠実度は相当に高い。私のようなアマちゃん格闘家には，オリジナルとの違いがわからないほどである。

登場キャラクターは全部で12人。元祖ストIIでは四天王と呼ばれる人物たちを選ぶことができない，ダッシュからはこいつらも使えるようになった。あまり思い出したくなかったが，ベガはちゃんと強い。

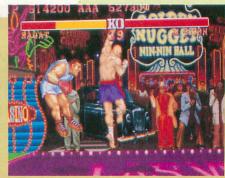
グラフィックは1ドットの違いもなく再現されている。これは恐らく，基板上のデータをそのままもってきているからであろう。家庭用ゲーム機ではドット数の都合でグラフィックの全面的な書き直しを余儀なくされていたが，それはX68000版において

X68000用　5"2HD版4枚組12,800円(税別)
カプコン　☎03(3340)0718

インドには象がちゃんと6匹いる

ヨガを究めれば，君もダルシムになれる!?

ブランカの伸び〜るパンチ

電撃ビリビリ，レントゲン写真

派手なネオンを背景にストリートファイト！

はまったく見られない。逆にディスプレイの解像度がゲームセンターより高いためか，オリジナルより美しいと錯覚するほどだ。画面がやや小さいのは，オプションメニューでCRT MODE 3を選び，ディスプレイの垂直振幅を調節すればなんとかなる。君のディスプレイにも，隆の流し目やサガットのニヤケ顔がクッキリと再現されることだろう。

そして，ゲーム性そのものに関わる部分での違いもまったくといっていいほどわからない。

たとえば，プレイヤーの操作の癖によってまれにキャラが思いもよらない行動にでてしまうことがある(ただし達人プレイヤーの突っ込みは却下)。「またやってしまった」という経験は誰にでもあるはずだ。そういったレバーやボタン入力の微妙なタイミングに起因するプレイ感覚までもが，ちゃんと移植されている。これには驚いた。

ということは負けても「これゲーセンとパターンが違うからしょうがないよ」などと苦し紛れのいいわけをすることが許されないわけだな。家にいながらも同じ苦渋を味わうことができてしまう。本当に，これは基板をもっている感覚に限りなく近いといえる。

内蔵音源によるサウンドは，完璧とまではいかないがまずまずのレベル。PCMが複数チャンネルであるということを除けばCPシステムとX68000の音源はほぼ同じスペックであるから，当然といえば当然か。

で，このソフトはなるべく基板の雰囲気をそのまま再現したいということから，4チャンネルのマルチAD PCMドライバを搭載。起動時にMPUを判別し，高速な68030ならばドラム音や効果音が途切れないという優れものだ(X68000でも一応使えるようだが)。やはり声と攻撃のヒット音が重なるとひと味違う。じっくり聞くと音声の重なりが悪く感じられるところもあるが，まずはめでたい。

ただ不満はいくつかある。まずドラムとFM音源，効果音の音量バランスが悪い点だ。攻撃のヒット音も多少違って聞こえる(防御の音ってこんなの？)。ここはなんとかしてほしかったところ。容量節約のためか，音声データのみ若干周波数を上げてサイズをケチっているようなのも，非常に気になる。ガイルの必殺技が「ソニックブー」なのはショック！

ボーナスステージも忠実移植

対戦者求む！

せっかく家で遊べるのだから，ターボモードや対戦モードなどの設定を充実させろとかいう意見もあるだろうが，余分なエッセンスを一切払い除け，業務用基板の完全再現のみに徹するという姿勢にも好感がもてる。ここは賛否両論あるだろうが，これぞ私の待ち望んでいたものだといいたい。

また，ストIIの醍醐味は対戦プレイでの人間同士の駆け引きにある。いや，「対戦こそ」が正しいかもしれない。ボタンとスティックが織り成す無限のバリエーション。究めたと思えばまだ上がいる。格闘家に休息などないのだ。

X68000版でも当然，ジョイスティックを2本つなげばゲームセンター状態になる(ぜひともアダプタをもうひとつ手に入れたいところ)。さらに贅沢をいうと，21インチくらいの大きなディスプレイでやりたい。これでストII対戦人口が5万人ほど増殖するのは間違いなし？　家庭用マシンじゃ不満だがゲーセンに行く金はなし，という人も，命枯れるまで対戦に燃えてくれ。家で腕を磨いておけば巷でヒーローになれるかもしれない。

誰か私のザンギエフに吸い込んでほしい人，いらっしゃーいってか。フンッ！

いよいよ戦いも終盤戦へ

こりゃええわ

RAMが多いとオンメモリ，メガドラのパッドがつながるなど親切な面も多い半面，X68000でも使えるはずのADPCM4.Xがデフォルトでは使えないとか，コンフィグモードが使いづらいなどの不満点もある。本文では触れなかったが，MIDIの音楽はまだまだ改善の余地あり。

ただ，まさかこれだけのものがX68000で遊べるとは思いもしなかっただけに，完璧に近い移植は夢のよう。X68000もまだ十分いけると再認識させられた。

あと，ハードディスクへのインストールに引っ掛かった人は今月号の質問箱を読むといいかもしれないぞ。

総合評価	0　5　10
移植度	★★★★★★★★★
サウンド	★★★★★★
グラフィック	★★★★★★★★★
格闘スピリッツ	★★★★★★★★★
操作性（6ボタン）	★★★★★★★★★
操作性（2ボタン）	★★★★★

餓狼伝説の続編が早くも登場!

Asakura Yuji
朝倉 祐二

現在,「餓狼伝説SPECIAL」が発売されるなど,人気の高さには定評がある「餓狼伝説」シリーズの2作目がX68000に登場! 4ボタンパッドも同梱されるし,NEO・GEO版にもなかった四天王が操作できる点もポイントが高いぞ。

「餓狼伝説2」は,SNKが送り出した対戦型格闘モノのゲームである。NEO・GEO版を制作したSNKでは以前にも対戦型格闘モノとして,「竜虎の拳」「餓狼伝説」を制作してきた。いずれの作品も人気はあったようである。だが,対戦型格闘モノといえば「ストリートファイターII」(以下ストII)が全盛の時代。猫も杓子も「ストII」に熱中していたのだから時期が悪かった。

さて,「餓狼伝説」の続編として発売された「餓狼伝説2」。美しいグラフィックに個性あふれる8人のキャラクター,ゴージャスなサウンド,豊富な効果音,戦いのフィールドに奥行きをもたせた2ラインバトル,各キャラにSNKお得意の,操作方法を秘密にした超必殺技,そして特に前作に比べて大きく変化した対戦プレイを念頭においたゲームシステムをひっさげて登場した。秋葉原ではNEO・GEOを2台店頭に並べて自由に対戦できるようにしていたところもあったが,これが大人気だった。なんたってゲーセンで100円払って遊ぶものとまったく同じクオリティのものがただで遊べるのだからそれも頷ける。そんなわけで「餓狼伝説2」は「ストIIターボ」人気が下火になった時期に,NEO・GEOにおいて空前の大ヒットゲームとなったわけである。

「餓狼伝説2」は4ボタンが基本である。2ボタンでは操作性が著しく損なわれてしまう恐れがあるが,X68000版には初回生産分に限り,4ボタンジョイパッドがもれなくつくそうである。NEO・GEOからの完全

製品版では四天王も加わる予定

移植。さらに対戦時は棒使いのビリー・カーンやクラウザーなどのNEO・GEO版ではCOMキャラだった四天王をプレイヤーが操作できるようになるそうだ。これで9,800円とは気前のいい価格設定だ。

対戦型格闘モノとは?

今回は「餓狼伝説2」のサンプル版が編集部に届いたので,これをもとにレポートしよう。その前に「餓狼伝説2」というより,対戦型格闘ゲーム自体をまったく知らない人(いないかもしれないが)のために説明をしておきたい。

対戦型格闘モノというのは,対戦相手を殴り,蹴り,さらにはキャラクター固有の必殺技を使ったりして相手のバイタリティ(体力)を減らしていくゲーム。バイタリティの減り方は受けたダメージの大きさに比例し,バイタリティが0になると戦闘意欲を失くしてその場に倒れ込んでしまう。その時点でバイタリティの残っているヤツが勝者であり,1ラウンドを取ったことになる。また,一定時間経過した時点で双方ともバイタリティが残っている場合は,バイタリティの多いほうが勝者となる。2ラウンド先取すれば勝ちとなり,次の対戦相手に勝負を挑むことができる。ちなみに「餓狼伝説2」では8人+四天王の12人のキャラクターが対戦相手である。

超必殺技を探せ!

では,いちばん重要と思われる操作性について話そう。サンプル版だということを頭に入れて読んでもらいたいが,16MHz以上のX68000,またはX68030ではNEO・GEO版とほぼ同じゲームスピードと操作性を実現している。マニアから見ればいろいろと違う点はあるかもしれないが,いずれにしろゲーム性を損なうほどのものではないから安心してほしい。

基本操作はスティックと弱パンチ,弱キック,強パンチ,強キックの4ボタン。2ボタンの場合はボタンを押す長さで弱と強を使い分けることになる。スティック操作とボタンの組み合わせで必殺技を繰り出す。さらに敵の攻撃を受けて体力ゲージが赤くなると,攻撃力の大きい超必殺技が使えるようになる。自在に出せるようになれば一発大逆転も狙えるすごいヤツだ。これがアツい。対COM戦ではCOMが超必殺技を使ってくることがある。特にテコンドーのキム・カッファンの超必殺技「鳳凰脚」は,いわゆる乱舞ですごくかっこいい。超必殺技の出し方については,おそらくマニュア

X68000用　5"2HD 6枚組9,800円(税込)
魔法株式会社　☎078(261)2790

モニュメントバレーを背にライダーキック

燃える日本男児,ジョー東

お詫び 12月号で「餓狼伝説2」の発売日が12月22日とありましたが,12月23日の誤りでした。関係者の方々にご迷惑をおかけしたことをお詫びいたします。

ルには書かれないのではないかと思うが，その筋の本にはすでに公開されている。でも自分で探すほうが楽しいだろう。

ちょっとヒントになってしまうけど，超必殺技のなかには，弱キックと強パンチ同時押しというのがある。こんなとき2ボタンを使って一方のボタンを短くして，一方のボタンを長く押すなんてのは，やろうと思ってもなかなかできないことだと思う。だから絶対4ボタンで遊んでほしい。

げえ〜，押し潰されるう

う〜む，煎餅じじいに挑発されてしまった

最大の屈辱技？　それは挑発だ！

見出しにも疑問符をつけたように，技ではないが対戦相手から受ける最大の屈辱は挑発だろう。「ストII」でもピヨッたときに投げにこないで，「フフン」と鼻笑いしながら小パンチを一発お見舞いしてくれる小生意気なガキがいるらしいけど，「餓狼伝説2」ではフフンの鼻笑いができない人の代わりに画面のキャラクターが挑発ポーズをとってくれるのだ。やり方は簡単で，敵との間合いを広くあけて強パンチ。たとえばテリー・ボガードの場合，顎を引いて上目づかいのポーズをとり右手で対戦相手に手招きで「かかってこいよ！」という素振りをしながら，「Hey, come on come on!」という。そして左手は帽子のつばをつかんでいて，帽子をかぶり直すという余裕の態度(突っ込まれる前にいっておくが,左向きのときは左右逆だ)。ピヨって身動きできない相手にこれをやると，相手に与える精神的ダメージは大きいぞ！　ほかにも挑発パターンはキャラクターごとに違うので面白い。十兵衛なんかは煎餅を食い出すし，舞は扇子を広げて「オホホホホ！」の高笑い。ちょっとでもチャンスがあったら，相手を挑発してしまうのが「餓狼伝説2」の楽しい遊び方，かもしれない。

2ラインバトルシステムは健在

前作の「餓狼伝説」にも備わっていた一風かわったシステム，それが2ラインバトルシステムである。これはキャラクターの移動に拡大縮小を組み合わせて画面に疑似的に奥行をもたせ，手前のライン，奥のラインというように戦いのフィールドを2つ用意したシステムだ。対戦相手が飛び道具で攻撃してきたら，「ストII」などでは防御するか，自分も飛び道具を出して相殺するか，ジャンプしてかわすしか選択肢がなかったが，「餓狼伝説2」ではこれにライン移動という選択肢が加わる。

自分が手前のラインにいるとしよう。対戦者が飛び道具で攻撃してきたときにライン移動をすばやく行えば，自分は奥のラインに移動し，まったくダメージを受けない。飛び道具を追いやったあと再びライン移動を行えば，手前のラインに戻ってくるといったようにである。対戦でも強いヤツはライン移動をうまく使っているようだ。

グラフィック，BGM，SEは？

わが家のNEO・GEOはビデオ出力でつないでいるせいかもしれないが，X68000版はNEO・GEO版を超えるのではないかというほど綺麗なグラフィックである。キャラクターも背景もである。ただし，なぜかアンディのステージだけは，川の向こう岸を走っているバイクのあたりがちょっと違うのだが……なぜだろうか？　ほかはNEO・GEO版に忠実に移植されている。

「餓狼伝説2」はBGMも評判がよく，X68000版でも音楽がどうなるかは気になるところだが，これについては現在まだサンプル版ということで評価はできない。MUSICドライバは前作と同じくZ-MUSICシステムを使っている。

魔法株式会社が移植をした前作の「餓狼伝説」でもMIDIに対応していたくらいだから，今回もしっかりとMIDI対応となっている。対応機種はSC-55，MT-32となっているのでNEO・GEO版の高音質をそのまま再現してくれるよう期待していよう。サンプル版にはFM音源版のデータのみが収録されていた。さすがにNEO・GEO版のサンプリング音源を使ったBGMに比較すると見劣りするが，これはしかたがないかもしれない。

BGMに続いてSEについて話そう。「餓狼伝説2」では実にたくさんの音声がサンプリングされている。ただのパンチやキックでも「アチョッー！」「オゥ！」「イヤッ！」や，勝ったときの「日本一ィ！」「オッケー！」「フゥッ！」などなど……。本当に種類が多い。この豊富なサンプリング音声がゲームを面白くしていることは間違いない。X68000はADPCMを搭載しているから，サンプリング音声もバッチリのはず。残念ながら今回のサンプル版では，ADPCMデータが不完全だったり，BGMとADPCMの音量のバランスが悪くて聴き取れなかったりしたのだが，その点は製品版では直されているだろうから心配ない。

なおグラフィックデータ，サンプリングデータはハードディスクにインストール可能である。ちなみに必要な空き容量は約8Mバイトだ。

製品版を待て！

あまりにも綺麗なグラフィックに驚いた，これがサンプル版を見た第一印象だ。操作してみるとちゃんと動く。これは期待できるぞ，というわけで私も製品版を楽しみに待っているひとりである。現状ではやや問題がある10MHz機でのゲームスピードも改善されるかもしれない。それに今回は4ボタンジョイパッドの到着が締め切りに間に合わず，使用感を報告することができなかったのが心残りだ。次号以降，必ず製品版の詳細報告をするから楽しみに待っていてくれ。

そうそう，ハードディスクを持っている人は空き容量を作っておくように。

グラフィックは本当にいい出来

クロービス，お見事!

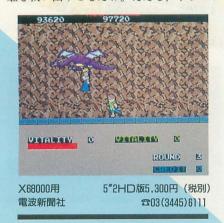
Yaegaki Nachi
八重垣 那智

剣と魔法とドラゴン，そしてお姫さまの登場するアクションゲーム「ドラゴンバスター」。往年の名作として，移植希望の多かった作品がアンソロジーシリーズに登場です。さあ，剣を振り振りドラゴン退治だ！

「リブルラブル」のレビューで書いたが，ナムコの黄金期と呼んでもいい時期というのは，「ギャラガ」から「パックランド」までの間である。その時代はゲームセンターのゲームといえばナムコであった。最近では，ポリゴンものかリメイクものしか出さないナムコしか知らない人にとって，そんな時代があったことすら信じられないかもしれない。

今回，すでにお馴染みとなった電波新聞社のビデオゲームアンソロジーシリーズの最新作として登場する「ドラゴンバスター」は，その余韻の残る1985年の初めにリリースされている。黄金期を過ぎたとはいえ，当時の最高峰メーカーであるナムコが送り出した，この「ドラゴンバスター」というゲームは，極めてエポックメイキングな作品であった。これを知らずしてゲームを語るということは，ロッテリアでマックシェイクを注文するくらい恥ずかしいことだといっても，過言ではないだろう。

入門「ドラゴンバスター」

まずは簡単なゲーム内容の説明から始めよう。「ドラゴンバスター」は，横視点のジャンプ型のアクションゲームである。プレイヤーである剣士クロービスの目標は唯ひとつ，セリア姫をさらったドラゴンを倒し，姫を救い出すことだけ。ただし，ドラゴンが潜むドラゴン山に向かう道には，数多くの難所と，そこを根城にするモンスターたちが，プレイヤーの行く手を阻んでいるので，それらを倒しながら宿敵であるドラゴンとの戦いに向かっていくという仕組みになっている。

X68000用　5"2HD版5,300円（税別）
電波新聞社　☎03(3445)6111

自分の道を進んでドラゴン山を目指せ

プレイヤーは，ジョイスティックの左右で移動し，上でジャンプして下で座る。ボタンで剣を振り，もうひとつのボタンでアイテムの巻物をファイヤーボールとして発射できる。ここまでは，実際にゲームをやってみなくても，簡単に予想がつく。問題なのはこのあとで，レバーは4方向で斜めはなし。斜めにジャンプするには，移動中にすばやくレバーを横から真上に入れ替えなくてはならないとなると，かなり特殊な操作のゲームだということがわかる。

それだけには留まらず，移動中に一瞬レバーを放し，もう一度進行方向に倒すとダッシュすることができる。ジャンプ中に，もう一度上に入れると，さらにそこから2段目のジャンプができるという，オーバーアクションも可能になっているということにいたっては，極めて独特の操作系をもつゲームなのは明らかだ。現在では，ダブルクリックでのダッシュや，2段ジャンプというのは珍しくないが，どれもこの「ドラゴンバスター」がルーツだといっても間違いではない。それだけこのゲームの独創性や影響はすごかったのである。

ドラゴンへの道

ゲームの流れは，ラウンドごとにドラゴン山に向けて全体マップの上のコースを選びながら進んでいくようになっている。どのコースを選んでも，目的にはなんら影響はないが，通過する各ポイントにおける，有利不利という違いは存在している。ポイントは山や廃虚・塔などで，そこに入ることでメインのアクションがスタートする。

各ポイントの内部は通路と部屋によって構成された垂直な平面のようになっている。その中を，プレイヤーを操作して出口を探すとそのポイントはクリアとなり，再び全体マップに戻るという流れになっている。

しかし，通路には各種雑魚モンスターがウロウロしているし，部屋の中に入ると，その部屋の番人であるルームガーダーと戦わなくてはいけない。しかも，脱出のための出口や，ファイヤーボール補給などの各種アイテムはすべてこのルームガーダーたちを倒すことで出現するので，プレイヤーはとにかく自慢の剣で敵を倒していかなければならないのである。

戦士はくじけない

プレイヤーはこうした戦闘で敵からダメージを受けると，バイタリティが減少する。このバイタリティがなくなると，即ゲームオーバーなので，慎重に敵と戦わなくては

基本中の基本，兜割りィ！

ならない。これも残機制の多かったアクションゲームにおいて，アナログ的な概念の導入された独特のものであった。とにかく一度死んだら終わりというのは，かなりのプレッシャーであった。一応，ポイントを抜けると多少は回復するし，アイテムの中には回復する薬や，バイタリティの上限をアップしてくれるキノコもあるので，それほど神経質になってプレイする必要はないのが，やや安心というところだ。

こうして苦難(?)の末にドラゴン山にたどりつけば，ついにドラゴンとの決戦。見事にドラゴンを倒すと1面クリア。面が変わって，次のドラゴン山に向けて終わりのない戦いが続いていくのだ。特定のステージでは，さらわれた姫が登場するが，助かろうと再びさらわれようと，ゲーム進行にはあまり関係ない。プレイヤーは黙々と，次のドラゴンを倒しにいくだけである。

この「ドラゴンバスター」は，ステージが進んでも特に目新しいものは登場しない。敵は強くなっていくが，その基本は，序盤からあまり変化しない。マップ上のポイントも同じようなものの繰り返しだ。

しかし，限られたものの組み合わせを変え，絶妙に配置し，自在にゲームの構成と演出を組み立てる手法は，当時のナムコが得意中の得意としていたものであった。このゲームも例に洩れず，全体マップの構成，各ポイント内部の構造，敵の設定というものまで凝りに凝っている。ゲームを進めていくうえでは，こういった細やかな設定を予習し戦略を練ることが重要になっている。とにかくプレイヤーは，ゲームを深く理解し，隠された作者の意図や仕掛けを楽しまなくてはならない。そこにテクニックがあり，それを駆使できる奥深さがあるのだ。

その指先を鍛えよ

ここで，ゲームを進めるためのいくつかのテクニックや設定を紹介していく。

まず，いちばん重要なのが「兜割り」で

スケルトンにはファイヤーボールが効かない

後ろから襲う陰険殺しも立派なテクニック

扉を開けて次のポイントに向かおう

ある。ジャンプ中に剣を振り，すかさずレバーを下に入れる。すると剣は水平に保持され，敵に倍のダメージを与えることができる。ルームガーダーに向かって斜めジャンプしながら，一撃で倒すというのは，もう当たり前のテクニックだ。これを2段ジャンプや，穴を落下中にやると「垂直斬り」になる。攻撃力は同じ。

次に「陰険殺し」である。ルームガーダーは自分の部屋が画面に入ったときに，プレイヤーの方向を向いて待ち構える。これを逆用し，一度画面に部屋を入れてから，画面の外に出さないように回り込むと，敵を背後から攻撃することができる。先制攻撃をかけられるので，敵が反応したときにはすでに勝負がついてしまう。比較的重要だができる場所は限られているので，マップを周りながら使える場所を覚えていくと上達の手掛かりになるハズだ。

また，ルームガーダーの出すアイテムも残さず活用したい。先に挙げたもののほかにも，魔法使いの空飛ぶ剣を無力化する盾や攻撃力倍増の剣がある。ファイヤーボールを補給してくれる巻物の一部には，回転して大ダメージを与えるスペシャルも混ざっている。これらは，各ポイントに固定で配置されており，全体マップでのルート選択が入手のカギになっている。コースによっては入手できないものもある。しかも，ポイントの中には，ジャンプを間違えて落ちると戻れないような部屋があり，欲しいアイテムは往々にして，そういった部屋にあったりする。つまり，計画性と実行力の双方が必要になってくるのである。

最後に，敵の特性の把握も忘れてはいけない。たとえばドラゴンだが，斬ったときに赤く光る部分は弱点で，そこを斬ると倍のダメージを与えることができる。さらに，「兜割り」やアイテムを組み合わせれば，一撃で最高32ポイントのダメージになる。ルームガーダーや雑魚も，剣の効かない敵やファイヤーボールの効かない敵，一時的に無敵になるものとさまざまだ。どれもこれも実際にゲームをプレイして覚えてほしい。それがすなわち，このゲームがうまくなるということだからだ。

戦いは朽ちない

おまけのようになってしまったが，移植の出来は上々である。各種パソコンに何度か移植されているので，ユーザーがどういった違いに敏感なのかわかっているのだろう。違いといっても，いまだにオリジナルをどれだけ覚えているかは疑問ではあるが。

ただ，アンソロジーのシリーズ全体に感じる，X680x0ならではの物足りなさは今回も気になった。贅沢をいってはいけないのかもしれないが，懐かしがって遊ぶだけではなく「いま，X680x0で遊ぶ」理由が欲しくなるのである。しかし，よいものを広め，いつまでも残すという意味では，今回も評価できるだろう。

斬る!!

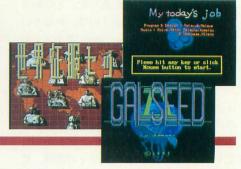
いまでこそ変化が地味で単調だが，当時はドラゴンも大きく見えたし，迫力満点のゲームであった。月日というのは無情なもの。そうそう，王冠と杖を揃えて姫を助けると，姫がレベルアップしていくので，男は剣や盾なんかに頼ってはいけない。しかし，やってみてわかったけれど，人間の記憶というのは実に不確かなもので，あれだけ毎日やっていたのに，ほとんどマップを覚えていないというのはショックだった。こんなに記憶がアテにならないのなら，借金や締め切りを忘れるのも仕方がないことなんだと納得できるというものだ。人間ていいなぁ，うんうん。

総合評価 (0 / 5 / 10)

項目	評価
ゲーム性	★★★★★★★★
技術	★★★★★★★
サウンド	★★★★★★
グラフィック	★★★★★★
懐かしさ度	★★★★★★★★★
バニー(笑)	★★★★★★★★

歓迎,とんがったゲーム御一行様

Kiyose Eisuke
清瀬 栄介

LOGiN主催のソフトウェアコンテスト受賞作品の12本を集めた「X68000傑作ゲーム選」。それぞれ個性的で，粒の揃ったゲームが収録されています。X68000のアマチュアパワーの集まったお勧めの1冊といえるでしょう。

先月号の特集に,「自分の好きなものを思いっきり作ろう。マーケットがあれば必ず認められる」と書いたら，さっそくその実例がやってきた。

この「X68000傑作ゲーム選」は，LOGiN誌が主催する「ログインソフトウェアコンテスト（略称ソフコン)」で優秀な成績をあげた投稿ゲームをまとめたものだ。シューティングにパズル，シミュレーション，果ては分類不可能なものまで12本が収められている。

アマチュアのソフトに「クオリティが低い」というイメージがつきまとっていたのは昔の話。特にユーザーの活動が盛んなX68000では，同人ソフトやPDSが市販のものよりクオリティが高いことも珍しくない。

ただ個人のセンスが色濃く出るだけに，市販ソフトではできないブレイクスルーを成し遂げる場合もあるが，好き嫌いの分かれるものになってしまうこともある。はたしてここに挙げた12本はどちらだろうか。

定石は強いのよ

●ガルシードII

まずはソフコン第1席を取った3本から見てみよう。最初はシューティングゲーム「ガルシードII」。はっきりいって，特に奇抜なところのないゲームだ。しかし，コンテストの第1席を取るだけあって，デザインやバランスの実力には素晴らしいものがある。全6面＋ファイナルステージの構成で，自機は3種類のパワーアップパターンを選べる。

完成度はピカイチの「ガルシードII」

最初はステージ構成も敵の攻撃ももの足りない感じがするが，後半からラストシーンにかけてどんどん盛り上がってくる。予備知識なしにプレイしてもクライマックスに近づいてくるのが感じられるが，これは相当な実力がないとできない技だ。ボスが本当に「敵役」として感じられる演出もなかなか凝っている。

難しさは「普通の人」コースだと，数回のチャレンジでクリアできるくらい。腕に覚えのないボクにはちょうどいいバランスだった。うまい人には「人間以外」コースなどの難易度がある。

ソフコン入賞作というから，もっと奇抜なゲームをイメージしていたのだが，オーソドックスで完成度が高かったのが意外だった。

お勧め度 ★★★★★★★★★★

これぞアマチュアのデザイン！

●世界征服セット

こいつはボクの期待どおり。とにかくオリジナリティとアクの強いゲームだ。「世界征服シミュレーション」という解説がついているが，ほとんど当てはまらない。あえていうなら，数字合わせゲームをシミュレーションっぽくアレンジしたような感じだ。

このゲームでは全世界がたくさんの国に分かれ，国ごとに数字がついている。プレイヤーは隣国から3,4,5というふうに連続した数字を拾い集め，攻撃を加える。相手の国力がゼロになったら国を占領することができる。数字はリアルタイムで変わるのでいかにすばやく数字を集めるかが勝負になる。モタモタしていると，自分の領土で反乱が起きてしまう。自分のいる国を奪われるとゲームオーバーだ。

この反乱，後半になると加速度的に増えてきてスリリングな展開が味わえる。頭を働かせながら指先を正確に動かすのが，昔のゲーム電卓のような感覚だ。

1プレイは5分ぐらいでカタがつくのだが，まさに「あと少し！」というところでゲームオーバーになるようにできていて，

まさにアイデアー発「世界征服セット」

X68000用　5"2HD版4枚組4,980円（税込）
アスキー出版局　☎03(5351)8194

気がつくとのめりこんでいる。面白くても人に説明できないゲームなんて，市販されるわけがない。アマチュアソフトでしか遊べないタイプの典型だ。

お勧め度 ★★★★★★★★★★

3Dゲーム素人風の迫力添え

●ブラディオン

ひとことでいってしまえばディスク1枚に収まった「スターブレード」＋「スタークルーザー」。

懐かしい雰囲気をもつ「ブラディオン」

画面に出てくる物体はすべて板で表現されている。ポリゴンよりさらに原始的。昔のATARI800を思い出してしまった。ただ，そのぶん動きはいい。最小限の努力で最大限の効果を狙うのは正しいアマチュアリズムだ。画面写真で見るとなんの形だかよくわからないかもしれないが，動いていると案外気にならない。

プレイヤーはマウスで照準を合わせて射ちまくる。自機の進むコースが決められたステージと，そうでないステージがあり，自動操縦では「スターブレード」のようにひたすら敵を撃墜し，手動操縦では「スタークルーザー」のように敵を探し出して撃墜するという楽しみがある。

解説によると演出もウリのひとつらしいのだが，ボクにはよく伝わらなかった。「こいつを倒せ」みたいなメッセージは出るが，ストーリー性があるとは思えない。BGMがないのも，演出不足と感じる原因のひとつだろう。

ただ，ひとりでここまで3Dシューティングを作り上げてしまう根性はやはりすごい。演出がわからないといっても，キチンとゲームになっているわけだし。魚雷が飛んでくるのにいたってはちょっと真似しすぎという気もするが，今は自分の作りたいものを作っている段階なのかもしれない。

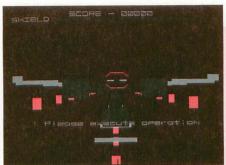
お勧め度 ★★★★★★☆☆☆☆

接待の席にどうぞ

●キャメロット

次は2席を取ったゲーム3本。

「キャメロット」は対戦専用のシミュレーションゲーム。同人ソフトなどでもよくみるタイプだ。

画面を見てもらえればわかるが，人間の形といい，戦闘の方式といい，あちこちが「ポピュラス」っぽい。お互いに3人の人間を操作して，ヨーイドンで画面上の宝箱を自分の陣地に回収にいかせる。

そのうち相手の陣地の宝箱の取り合いが始まるので，戦闘があったり，番をしたり，あるいは人間の代わりに戦ってくれる魔法石を作って配置したりして，だんだんと運動会の棒倒しみたいになってくる。

偶然性に頼るアイテムがないので，純粋に戦略で決着がつくということだが，そこらへんの駆け引きがわかる前にテストプレイが終わってしまった（なにせ12本もあるもんで）。作者の方，ごめんなさい。

しかし，やっぱり友人がきたときに対戦ゲームがあるとありがたい。他人がプレイしているのを後ろから眺めているのはあんまり面白くないしね。だが，1P専用がないと同レベルの友人を作るのがけっこう難しそう。やっぱり対戦ゲームの思考ルーチンを作るのは面倒なのだろうか。ちょっと残念な気もする。

お勧め度 ★★★★★★☆☆☆☆

ストレスのあとの爽快感って

●紫陽花

ビットというつぶつぶを4つ並べて消すという，ルールだけ聞くとよくありそうに聞こえるパズルゲーム。違うのはトラックボールを使った指先の感覚が勝負になるところ。

宙をさまよっているビットをトラックボールで操作し，うまくくっつけてやる。すべてのビットを消せばクリア。普通にやっていてもクリアはできるが，ステージの配置をよく見ると，爆弾の導火線のように「この色を揃えると一気に終わる」というところがある。それを見抜いてクリアまでのタイムを競うのが正しい「紫陽花」の遊び方だ。うまく読んでいっぺんにビットが散らばるさまはなかなか爽快。

だけど，これが結局パズルなのかどうか疑問。タイムを競うゲームなのに，欲しいビットがあるかどうかは偶然だし，頭ではどこにビットをつけるのかわかっていても，指先が鈍いとヘンなところにビットがついてしまう。遊んでいるとなんだかストレスが溜まる。

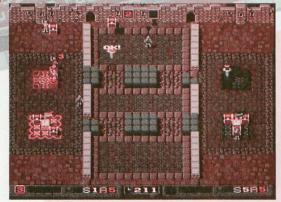
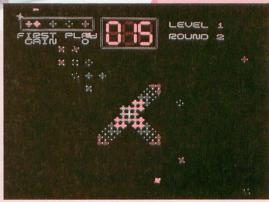
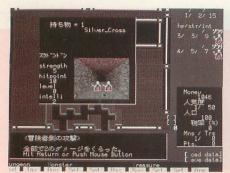
これがクリアできるようになったら，あなたもハシで蠅がつかめるようになるかもしれない。

お勧め度 ★★★★★★☆☆☆☆

逆転の発想よ

●ダンジョンマネージメント

魔王になって迷宮を支配するというシミュレーションゲームだ。こういった逆転の発想は「Wizardry4」を始め，あちこちから出ている。だがこの「ダンジョンマネージメント」は，迷宮を「経営する」という点にオリジナリティが感じられる。

まず宝物を買ってきて客寄せ（？）にし，

対戦専用がちょっと残念な「キャメロット」

ストレスと爽快感がいっぱいの「紫陽花」

営業も楽じゃない「ダンジョンマネージメント」

ついつい手が出る「レーシーズ」

怪物を雇って番をさせる。やってきた冒険者を倒して収入を得るわけだ。迷宮もこんなふうにデザインされると，まるでテーマパークである。

その収入で魔王の軍隊（迷宮にいる怪物とは別）を雇い，自分と同じようにダンジョンを経営している別の魔王と対決する。目指すは全ダンジョンの制圧となる。

この展開にはビックリ。立場を逆にするという発想はよくあるが，そこからまさか「信長の野望」みたいな「国盗り」ものになってしまうとは。次のダンジョンを手に入れてしまうと，前のダンジョンがどうでもよくなってしまう点と，ほかの魔王との対決の部分にボリュームがないのがややもの足りないが，アマチュアらしいインパクトのあるシミュレーションゲームである。ボクは続編を作ってほしい。

お勧め度　★★★★★★★☆☆☆

パズルはアマチュアの基本です

●レーシーズ

ここからあとの6本は過去のソフコン入選作である。

この「レーシーズ」も「紫陽花」と同じパズルゲームだ。一定時間すると画面上にパネルが出現する。小人を動かしてパネルを押したり引いたりして並べ替え，3枚同じマークか色を揃えて消滅させる。並べても消えないブロックや，それを消すためのブロックなど，ゲームデザインとしてのスパイスがちゃんと効かせてあり，クオリティは高い。対戦プレイ，協力プレイもあって，押さえるところはしっかり押さえている。

プレイにヤマができにくいのと，華麗なプレイ，ハデな展開というのに欠けるような気がするが，「ぷよぷよ」など最近ハヤリのタイプのパズルゲームに慣れているからそう見えるだけかもしれない。

また，この「レーシーズ」をアレンジした「キーパー」が，X68000版とアーケード版で発売される予定だ（X68000版は12月発売）。

お勧め度　★★★★★★★☆☆☆

こういうのを待ってました

●マイ　トゥデイズ　ジョブ

ほのぼの型アクションゲーム。最近の刺激が強く短時間で終わるアクションゲームの風潮に背を向けた作品だ。

ステージごとにミッションがあって，これをクリアするたびに，エピソードが展開していくというパターン。このミッションが「風船を集める」だったり「空飛ぶオバケの写真を撮る」だったりして，なんともほのぼのしている。

主人公が乗るのは人力飛行機。マウスを一生懸命転がして空にはばたくのである。弾よけもパワーアップもなし。ただ人力飛行機をコントロールする腕だけが試される。障害物に当たっても失速するだけで，地上に激突しないかぎりゲームオーバーにはならない。

最近はアーケードゲーム＝いいアクションゲームになりつつあるが，ハデでプレイ時間の短いゲームだけがアクションではない。家でしかできないアクションというのもあるはずで，この作品はそれをうまく表しているといえるだろう。

プログラム，グラフィック面は写真のとおりのクオリティの高さ。こういう市販ソフトで出にくいゲームが収録されていてこそ，アマチュアのゲーム選集を買った意味がある。ボクはこれがいちばん気に入った。

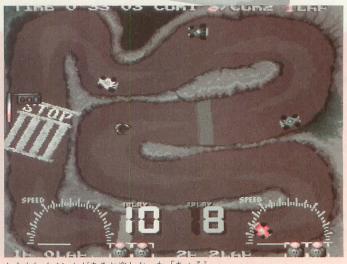
お勧め度　★★★★★★★★★★

ミニサーキットな気分

●ちょろ²

ちょっと前のアーケードゲームにもあった，レーシングゲームだ。1画面というのがやや貧弱だが，そのぶん信号やら橋やらいろいろな仕掛けが用意されている。夜のレースや雨や雪（！）の中の対戦まであるのだ。

ルールがわかりやすくて，対戦向き。逆にいえばひとりで遊んでいると，どこに燃えていいのかよくわからない。ひとりで遊ぶプレイヤーに与える動機づけがちょっと乏しいかなという気がする。

「マイ　トゥデイズ　ジョブ」は雰囲気作りがうまい

もう少しイベントがあると楽しかった「ちょろ²」

コンピュータも一緒に走ってくれるのだが，ヘタな車とうまい車の両極端。コースが短いのですぐ周回遅れになってしまい，デッドヒートはやりにくい。

プログラム的な品質は高くて，タイトル画面はもちろん，ゲーム画面にもチープな感じはないのだが，せっかくならそのセンスをゲームデザインにも生かしてほしかった。せっかく10コースあるんだからチャンピオンシップ戦でも作ったりすればよかったのに。

お勧め度　★★★★★☆☆☆☆☆

アイデアだけであまり面白くない「クアルト」

もう少し光るものがほしい「オーバードライバー」

名作にいたる道だ

●ラージス

始めに紹介した「ガルシードII」と同じ作者が作ったシューティングゲーム。「ガルシードII」はこの「ラージス」の続編という位置づけだ。

このゲームには独自のウリがある。捕獲ビームで敵をオプションとして使えるというのがそれ。ビジュアルシーンによるストーリー展開もあって，ステージによっては味方機が登場して一緒に戦ってくれたりする。

やはりゲームの完成度は「ガルシードII」に比べてだいぶ落ちる。せっかくのアイデアも，ゲームの展開のなかでこれを有効に生かせる機会はあまりない。

ビジュアルシーンの会話はかなり恥ずかしくなるものがあるが，若気のいたりということで見逃してあげよう。

お勧め度　★★★☆☆☆☆☆☆☆

ゲームにはアイデアが大事?

●クアルト

精霊と動植物が共存する世界を舞台にしたシューティングゲーム。どういう世界なのやら。どれが精霊かというとオプションに見えるRGBの丸い奴がそれ。

シューティングの場合は，Bボタンの使い道がアイディアの出しどころ。この「クアルト」の場合はBボタンで自機の大きさが変わる。弾を避けたいときは小さく，ガンガンいくときは大きくと使い分けられるわけだ。

しかし，さすがにこれはアイデア倒れという匂いがしなくもない。とっさに大きさを変えるなんて普通できないし，自分に向かって飛んできた弾は大きさを変えたぐらいではよけられないような気がする。新しい発想は，消化して自分のものにするのが大変なのだ。

お勧め度　★★★☆☆☆☆☆☆☆

ヘンな世界観とヘンな操作

●オーバードライバー

これまたBボタンの使い道にアイデアを注ぎ込んだシューティング。今度はヨーヨーの要領でオプションが飛んでいく。しかもこれがぶつかったものにくっついてしまい，自機とオプションの間の線に触った敵を倒すことができるという趣向だ。

ゲームが始まると，ネズミの大群が襲ってきたり，自機にキノコが生えたりして，いったいこの世界はどうなっているんだという感じ。テクニックがあるわりにセンスがヘンで楽しい。

だけど，この「オーバードライバー」もオプションをうまく使わせるようなデザインになっているかどうかとなるとちょっと怪しい。グラフィックもよく描けているので遊んでいて興醒めすることはないのだが，「燃える」と思うまでにはもう一歩か。

お勧め度　★★★★★★☆☆☆☆

お買い得なソフト集だ

1席を取ったソフトから昔のソフトまで収録しているため，作品ごとに多少レベルのバラつきがあるのは確かだが，12本はそれぞれの個性があって楽しめる。

全般的にハードゲーマーでないと遊べないようなゲームがないのが嬉しい。むしろ，ゲームに馴染みの薄い人たちにとって遊びやすいものが多いようだ。友達や彼女と遊ぶネタになるようなゲームもあって市販のゲームとはまた違った使い道があるソフト集である。

ゲームはすべて一般のファイルと同じように納めてあるから，ハードディスクにインストールすることも，ゲーム作りの参考にすることもできるだろう（ただし法的には著作権は作者に属する）。

アマチュアのソフト集ということで，クオリティに少し不安を抱いていたが，思いのほか楽しめるソフトが多かった。これで4,980円は十分お買い得だといえる。

さあ，あなたもアマチュアならではのとんがったゲームや風変りなゲームを試してみよう。これらのゲームを土台にさらなるインスピレーションが湧いてきて，あなたも新しいゲームを作りたくなってくるかも

出来としてはそれなりの「ラージス」

こういう活動は歓迎したいね

こういったゲーム誌のソフトウェアコンテストの存在は広く知られているが，その作品の内容などに十分な関心が寄せられているとはいえない。TAKERUでの供給もされているようだが，これもキャッチーな特定の作品しか話題を呼ばないという悩みがあった。

だから，こういったオムニバス方式の書籍による作品の供給は，ユーザーの活動を紹介し支援する方法のひとつとして評価できる。作品はディスク4枚に圧縮された形でついているが，ユーティリティをつけて初心者への心くばりもきちんとしてある。お目当ての作品のほかに，思わぬ収穫に出合うのも楽しいものだ。こういった努力を続けてほしいものである。

総合評価	0 … 5 … 10
お買得度	★★★★★★★★
マニュアル	★★★★★★
意外性	★★★★★★★
総合熱中度	★★★★★★★

1993年12月号のハガキ集計ベスト10 最近買って気に入ったソフトは？

POINT	タイトル	発売元	発売日
95	悪魔城ドラキュラ	コナミ	'93/7/23
52	ネメシス'90改	SPS	'93/11/12
44	ぶたさん	電波新聞社	'93/10/29
44	コットン	EAビクター	'93/9/24
20	SX-WINDOW ver.3.0	シャープ	'93/3/30
18	スーパーリアル麻雀PⅡ&PⅢ	ビング	'93/10/23
18	餓狼伝説	魔法株式会社	'93/7/23
16	MATIER Ver.2.0	サンワード	'93/10/20
15	ロボットコンストラクションR.C.	エレクトリックシープ	'93/7/30
15	クレイジークライマー/クレイジークライマー2	電波新聞社	'93/8/27
15	リブルラブル	電波新聞社	'93/6/25

（無作為抽出した1000通のハガキを集計）

先月号はページの都合でこの集計はお休みさせていただいた。と，いうことで前回の集計から2カ月が過ぎたわけだが，時の流れは順位の変動にきっちりと反映されている。

2カ月のあいだに発売されたソフトが4本新登場となった。「ネメシス'90改」「ぶたさん」「スーパーリアル麻雀PⅡ&PⅢ」そして「MATIER Ver.2.0」だ。

じらされ続けた「ネメシス'90改」は堂々の2位で，前評判どおりの満足度のようである。「待っていたかいがあった」とのコメントも。

そしてビデオゲームアンソロジーシリーズ6作目の「ぶたさん」もいきなり3位となった。画面を走りまわるかわいいぶたさんたちの爆弾バトルゲームである。同点3位の「コットン」も，先々月号で発売わずか数日後の集計にもかかわらずランキング入りしたものだが，その後届くハガキのコメントからも，ユーザーのココロを着実にとらえているのが見てとれる。

3人娘をムイていく麻雀ゲーム「スーパーリアル麻雀PⅡ&PⅢ」も支持を集めて6位。

バージョンアップした「MATIER Ver.2.0」は，10月号まで安定した人気を保っていた前バージョンと交代するかのように8位の座についた。11月号で紹介した付属の3Dステレオグラム生成ツール「FLICKER」への感想も寄せられている。

ここのところ，前評判の高い人気ソフトの発売が相次いだせいか，人気が分散し，それぞれのソフトごとの獲得ポイント数はやや下がっている。発売開始以来ぶっちぎり状態を続けている「悪魔城ドラキュラ」も今月は惜しくも3桁のポイント数は維持できなかった。しかしそれでも2位以下に大差をつけており，安定した1位の座を確保している。

そして，1993年6月号以来，毎月ランキング入りという根強い人気を誇っていたアクションゲーム「エトワールプリンセス」がとうとう姿を消してしまった。

さて，12月号のハガキでは，23ページの「SOFTWARE INFORMATION」で集計している「期待するソフト」のほうに挙げていた人が多かった「ストリートファイターⅡダッシュ」。これも発売後1～2日で，すでにランキング入りあと一歩という勢いだ。みんなが遊びこんだあとの来月の集計が楽しみである。順位の変動は？　ポイント獲得の点数は？　「悪魔城ドラキュラ」の作った記録は果たして破られるのか？

ウワサのソフトウェア(海外編)

COMBAT AIR PATROL

このコーナーも長いことご無沙汰していたが，久々に3D野郎を刺激するAMIGA用フライトシミュレータが出たので紹介しよう。昨年の暮れ頃にデモバージョンが出回って以来，情報が途切れたので心配していたが，無事に発売された。

■F-14での飛行を楽しむ

なんといってもF-14戦闘機で飛べるのが魅力(F-18も用意されている)。映画「トップガン」と「アフターバーナー」に採用されたこの戦闘機は，機能や性能といったものを超越したステータスをもって私のなかに君臨しているのだ。

戦闘機の動作が凝っていて，F-14の場合，低速飛行時は旋回性を上げるために主翼が前進し高速飛行時には主翼が後退するのだが，そんなところまで再現している。

空母からの発進も秀逸。映画を観た方はご存じだろう。あれが再現されているのだ。空母の甲板は狭いので，十分な助走距離が取れない。そこでカタパルトに機体を固定したうえでエンジン全開にしておき，リリースする。かっこいい。

■高いフライトシミュレータ技術

戦闘機のモデリングは異常の領域に入っている。主翼の下のミサイルの尾翼が1枚1枚モデリングされているほどの精密さである。パソコンでここまでやるか？　というくらいの凝りようである。

にもかかわらず高速で動作するのはすごい。地上物もけっこう作り込んであるし，相当遠くまで見えるのに，ノーマルのAMIGAでそこそこのスピード，速いマシンでやれば実に滑らか。層の厚いAMIGA用フライトシミュレータでもトップクラスといっていいだろう。動きが滑らかで，飛んでいて気持ちいいのだ。

そして時間管理するという機能は，実はAMIGAのフライトシミュレータでも珍しい。これにより飛行速度はどの計算機でも変わらず，速い計算機ほど滑らかに動くということが可能になっている。

フライトシミュレータ技術の集大成という雰囲気がある。とにかく思いつくフィーチャーをすべて入れてあるという感じ。

■退屈させないゲーム構成

フライトシミュレータ技術だけでなく，ゲームの作りもよい。構成自体は基本的に通常のフライトシミュレータと変わることはない。単発の任務やキャンペーンと呼ばれる一連の任務，訓練モードや対戦モードなどいくつかの遊び方がある。が，それだけでなく，退屈させない工夫が見られるのには好感がもてる。いいフライトシミュレータは飛ぶだけで楽しいということを知っていて，飛ぶことの楽しさをきちんと前面に出しているのだ。

たとえばインスタント・フライト。任務の説明とか武器の選択とかいった，うざったい作業をせずとも，とりあえず飛べる。

たとえば攻撃目標付近からのスタート。フライトシミュレータでは，基地から飛び立って目標まで数分間ぼんやりと飛び続けるということが起こりがちなのだが，そこをすっとばして目標の近くにいきなり出現することが可能。目標が遠くにある場合などに特にうれしい。

また，基地から発進しても，最高16倍速までの早送りが可能なので，途中のかったるい飛行を短縮できる。

■おわりに

販売元は「シャドー・オブ・ザ・ビースト」シリーズのPsygnosis。ちょっと意外に思われるが，最近はそうでもなくなっている。昔は「オープニングデモはすごいが，ゲーム性は最低なアクションゲームの販売元」という風評を欲しいままにしていたのだが，「レミングス」以来，ごくふつうのソフトウェア販売会社になってきている。そしてそれとともに，オープニングデモがぱっとしなくなってきてしまった。この「COMBAT AIR PATROL」のオープニングデモにしても，よくできてはいるが，最近のCD-ROM搭載機のものに比べるとどうしても見劣りする。少なくともオープニングデモのために買うという気はしない。本体がきちんとしているというのはいいことだが，AMIGAの怪しい魅力のひとつが失われていくようでちょっと残念。　(A.T.)

販売元：Psygnosis

EPA2 補講(その2)

プロジェクトチームDōGA
かまた ゆたか

先月はEPA2の使い方を解説しました。今月はいよいよ「森山効果」を分析します。この技法をマスターすれば，あなたのCGA作品の表現力は，確実にレベルアップするでしょう。しかし，予想以上に奥が深い……。

はじめに

前回のCGAコンテストで作品賞を受賞された客野優さんにお子さんが生まれました。名前は「伶太」くんだそうです。おめでとうございます。

当方からも，お祝いの品として「LEGOブロック」をお贈りしました。これはモデリングの基礎をたたき込むのが目的です。その後も誕生日ごとに，メタボールを理解するための粘土，続いて，レイトレーシングの原理を学ぶ針穴写真機，標準人体モデルを動かすためのミクロマン(もう売ってない？)などを贈ることを検討しています。目指せ，CGA英才教育！ DōGAオリジナル「大リーグマウス養成ギプス」開発中！

＊　　＊　　＊

さて前回は，EPA2の使い方を勉強しました。透過光やぼかし，そしてマクロ機能などはちゃんと使えるようになったでしょうか。

今回のテーマの「森山効果」というのは，「EPA2ビデオマニュアル」や「第5回CGAコンテストオープニング」など森山昇一氏(通称：宇宙人森山)の作品のなかで盛んに使われている手法です。爆発するカットなどで，放電の稲妻をピカピカと派手に光らせるものです。なにしろ迫力がありますから，ぜひ修得しておきましょう。

先日も，ある上映会で，

「稲妻バリバリの効果はどうやるのですか」

という質問を受けました。

「それは，今度のOh!Xのネタですので，そちらをご覧ください」

とその場はごまかしたのですが，後日またその方からお手紙をいただきました。

「稲妻の効果の方法がわかりました。EPA2でしょ」

確かにそのとおりなんですが，それがわかっただけでできるほど底の浅いものではありません。ということで，今回は，森山さんが蓄積してきたノウハウを惜しげもなく公開していきます。

このテクニックには，いったいどんな秘密が隠されているのでしょうか。では，実際にやってみながら，そのあたりを探ってみるとしましょう。

まずは準備

まず，適当な動画を用意してください。宇宙戦闘機でも作って，それが画面中央でゆっくり回転しながら，視点から遠ざかっていくようなのがやりやすいでしょう。私が用意したのは，写真Aの30フレームのカットです。こいつにビームが当たって爆発するように手を加えます。

EPA2を起動する前に，TEST1＊.PICをコピーします。

COPY TEST1＊.PIC TEST2＊.PIC

これは，EPA2では入力画像にオーバーライトしてしまうからです(最新バージョンでは，入力画像と出力画像を別々に指定できる)。ちゃんと元絵を残しておかないと，もう一度作画からやり直すはめになりかねません。

準備ができたら，

EPA2 TEST2001.PIC

でEPA2を起動します。

実際に稲妻を描き込む前に，まず「Sys.」に入って「Animate Mode」を「ON」にするのを忘れないでください。このへんの操作がわからない方は，先月号の復習が必要です。

ビーム光線

EPA2の使い方の復習がてら，とりあえずビーム光線が貫通するところを描きましょう。私は，1発目のビームが4フレーム目に，2発目のビームが10フレーム目に飛んできて，この2発目が当たるということにしました。まず，「File」の「Next」を3回クリックして，4フレーム目の画像を出します。かなりのアップですので，ビームは1フレームだけ思い切って太く描きます(写真B)。

このように極端に太い(ある程度面積がある)線を描く場合には，「Poly」が便利でしょう。「Poly」をクリックすると，「Free」と同じメニューが出ます。この「Poly」は，連続する折れ線を描き，その折れ線に囲まれた部分を塗りつぶすという機能です。難しく考える必要はあり

ません。試しにマウスカーソルを動かしてクリック，また動かしてクリックしてみてください。画面上に点滅する折れ線が伸びてくるでしょう。そして最後にダブルクリックすると，塗りつぶします。簡単ですね。とりあえず，右クリックでメニューを出し，「Undo」で試し描きを消してください。

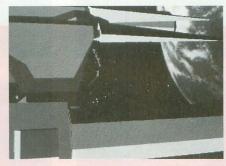
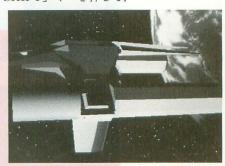
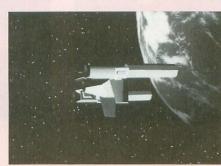
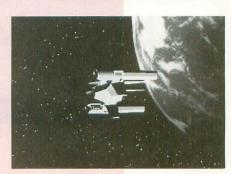
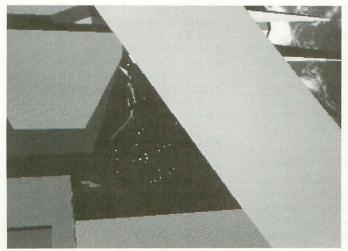
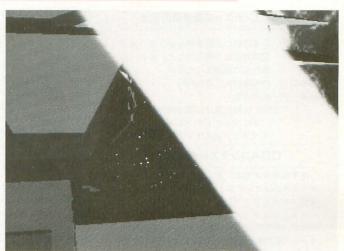
少し問題になるのが色ですが，RGBの各値を5～25にしておくのがコツです。たとえば，赤8，緑25，青12などにしてはいかがでしょうか。私は，この色で思い切って写真B［test2004］のように描いてみました。

次に，この色を光らせます。メインメニューに戻り，「Filter」，「Snow Fire」に入って，このビーム光線のところをクリックします。これでこのフレームは完成です(写真C)。「File」の「(Save and)Next」でセーブします。

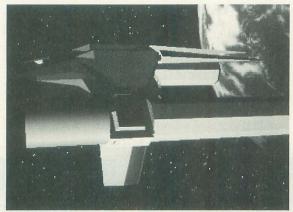
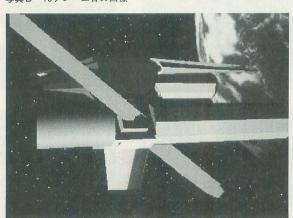
さて，次は2発目のビームです。まず，再び「Next」を5回クリックして，test2010.pic(写真D)を出します。このビームも同じような描き方でよいのですが，ビームが下から上に向かって撃たれたのがわかるように，今度は10フレーム目と11フレーム目の2フレームにわたって描いてみましょう。

最初のフレームのtest2010.picは，写真Eのような感じでよいでしょう。注意する点としては，4フレーム目のビーム光線と方向を揃えるためには線の太さなどに気をつける必要があるということです。この例では下から上へと撃っているのですから，少なくとも下のほうが幅が狭くないといけません。

このような直線を描く場合は，「Line」が便利でしょう。「Pen」の「<－－>」や「Shape」で筆を調節してください。

2枚目のフレームのtest2011.picは，写真Fのようになります。これを描くときは，「SHIFT」キーを押して，

写真A　だんだん遠ざかっていく宇宙戦闘機のカット（全30カットのうちの6カット）

写真B　4フレーム目の画像に1発目のビーム光を描き込む

写真C　写真Bのビーム光線を光らせる

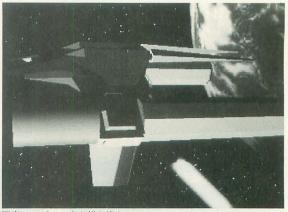
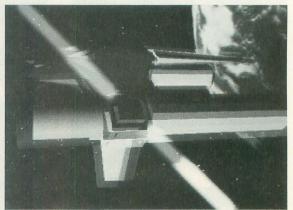
前のフレームの画像を透かしてみて(写真G)，必ずビームが同一直線上になるように注意してください。画面全体をひとつの直線で描くのではなく，機体の下の部分と，機体の途中から上の部分の2つに分けて描くと，貫通している感じが出ます。

光らせる処理もちゃんと行えば，これでビーム光線は出来上がりです。EPA2の使い方をだいたい思い出していただけたでしょうか。ここで，ちゃんとセーブしたのを確認してから終了して，一度アニメーションさせてみましょう。

```
mktch test2
hanim /m2 test2
```

「/m2」は，色数を6万5千色にするオプションです。一般に，透過光を使うと色数が増えてしまいます。このオプションを使用するとメモリ効率が悪いのですが，まぁ，短いカットですから，メモリが少ない方でも問題ないで

写真D　10フレーム目の画像

写真E　ビーム光を描き込む

写真F　11フレーム目

写真G　前のフレームの画像を透かしてみるとよい

▶CGAコンテスト事務局より◀

CGAコンテスト応募要項再確認

応募方法　応募用紙に必要事項を記入のうえ下記の住所まで郵送のこと。なお，応募作品の返却は行わない。
締め切り　1993年12月31日(必着)
応募・問い合わせ先
〒533 大阪市東淀川区淡路5-17-2-102
プロジェクトチームDōGA
「アマチュアCGAコンテスト事務局」

CGAコンテスト会場決定

毎年来場者が増え，寒いなか朝早くから行列ができるCGAコンテストだが，今年もさらに会場がスケールアップした。

さぁ，いまからカレンダーに印をつけよう。CGAコンテストは1994年3月6日(日曜)三宅坂ホール(地下鉄永田町駅下車徒歩4分)に決定だ。社会党本部がある建物だが，社会党支持者しか入れないなんてことはない。民主共和党支持者だろうが，甘党だろうが関係ないからな。

さらにビッグニュース。今年は大阪会場が変わる！　毎年，大阪会場は手抜きというか，J＆Pの教室で作品を上映していたにすぎない。しかし，今回からは東京での上映会に負けないイベントにしようと準備をしている。近畿地方に住んでいる人は，4月2日(土曜)に印をつけるんだ。場所は，摂津市市民文化ホール，JR東海道線千里丘駅から南に徒歩15分だ。

CGAコンテスト応募状況

気になる応募状況だが，毎年締め切り直前の10日間に集中するので，現在のところまだわからない。ただ，今年は募集の告知を掲載する雑誌などを増やしたこともあって，新人からの応募用紙の取り寄せが非常に多くなっている。たいへんよいことだ。

常連の方々の動きのほうも，いろいろな噂が入ってきている。まず，グランプリを2度も受賞した「SWORD」の森山さんだが，今年は1カット部門に出品するという，実にふざけた発言をしている。許SCSI！　逆に「猿蟹合戦」の宍戸さんは，グランプリへの返り咲きをねらっているとか，横浜動画クラブとしてかなり力を入れて制作しているらしい。昨年，「面会」「ハッピーバレンタイン」と2作を出品した客野さんが産休なのは，まぁ仕方があるまい。「TORNADO」の文月さんは，さすがに「TORNADO」シリーズはやめて，ちょっとした作品を完成している。ただ，著作権などの問題で参考出品になるかも。「解像連続体」の宍戸さんも元気で，いままでとは異なる作品を，間に合えば2作出すらしい。

う〜む，なかなか今年も常連陣健在という感じだ。新人も負けるんじゃないぞ！

しょう。

アニメーションをご覧になっていかがですか？　単に迫力が出たというだけでなく，ただ飛んでいるというデモが，ストーリーを持った作品のなかの1カットのようになった気がしませんか？　12フレーム以降の後半がちょっと寂しいですが，これからガンガン描き加えていきましょう。

森山効果の与え方

森山効果の与え方をひと言で解説すると，EPA2でアニメートモードの状態にして1フレームずつ稲妻を手描きし，マクロ機能を使って光らせるという作業になります。特に問題はなさそうですね。

では，引き続いて実際にやってみましょう。ビームが貫通した直後の12フレーム目からです。「Free」に入って，ビーム光線と同じ色の細い筆を作ります。そして「SHIFT」キーを押してビームが貫通した場所を確認しながら，その付近に描き加えます。最初は，貫通した部分に穴が開いている程度でよいでしょう(写真H)。そして「(Save and)Next」とします。13フレーム目は，同様にちょっと大きくします。

14フレーム目からは，穴を中心に放電現象が起こります(写真I)。小規模の放電現象は，物体のラインに沿って走らせるとよいでしょう。1枚1枚手で描くのは面倒ですし，あとで光らせたらそれなりに迫力が出るでしょうから，ていねいに描く必要はありません。むしろラフに描いたほうが迫力も出るでしょう。

17フレーム目ぐらいから，大きな稲妻が出始めます(写真J)。あとは，それをだんだん派手に大きくしていけばよいわけです(写真K)。

以上の作業が最終フレームまで終わったら，「File」の「Prev」で12フレーム目まで戻ります。そして，メインメニューの「Macro」でマクロ機能を使い，「Filter」「Snow Fire」で光らせて，「File」の「(Save and)Next」を選んで，マクロの定義を終えます。「Macro」の「Exec Repeat」でマクロを実行すれば完成です。

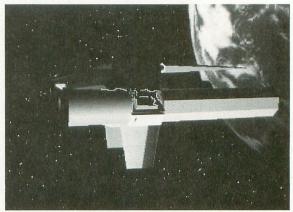
どうです。皆さんも完成した画像をアニメーションさせてみてください。むちゃくちゃカッコイイ爆発シーンが出来上が……？　なんだ，こりゃ！　カッコイイどころか，ぜんぜん爆発しているようには見えないぞ！　なぜだ？　何がいけないんだー！

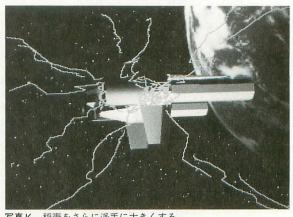
家元登場

森山(以下**森**)：アマイ！　甘いですね，かまたさん。

かまた(以下**か**)：あっ，これはこれは森山さん。どうしてこんなところに。

森：何いっているんですか。さっきからずっと，うしろで見てましたよ。

写真H　12フレーム目でビームが貫通して穴が開く

写真I　14フレーム目から放電現象が始まる

写真J　派手な稲妻が出始める17フレーム目

写真K　稲妻をさらに派手に大きくする

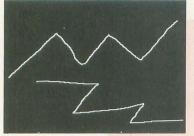
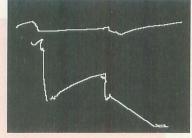
か：うっ。もっ，森山効果がうまくできないんです。どうしたらよいのでしょう？

森：森山効果への道は，一日にして成らずじゃ。努力と根性！

か：あのぉ～，それじゃ原稿にならないんですが……。

森：ウソウソ，ちゃんとコツがあるんですよ。ところでなんでもええけど，この「森山効果」って言い方，恥ずかしいんですけど。

か：じゃぁ，なんてしませう。

森：アニメ風透過光エフェクト。略してアニエフェ。

か：まぁ，本人がそういうなら「アニエフェ」でもいいけど，今後もアマチュアCGA界では「森山効果」って呼ばれるような気がするな。んで，私のアニエフェのどこがいけないんでしょうか。

森：う～ん，どういうたらええんかな。まず，ひと言でいうと，緊張感が足りない。

か：もう少し具体的にお願いしたいのですが……。

森：そうやね。いろいろあるけど，もうちょっと変化をつけなきゃいけないんです。

写真L－1

写真L－2

変化をつける3つのポイント

森：たとえば，線を１本描くにしても，写真L－1のように，みな同じような長さで折れ曲がるのではなく，写真L－2のようにサーッと長いところもあれば，短くグチャグチャしているところもあるというように……。

か：あっ，ほんとだ。カッコイイや。

森：それから次に，太さにも変化をつけてやらないと。かまたさんが描いたのって，全部同じ太さの線でしょ。

夫婦でQ&A

うさ子：もうすぐお正月ですね。私たちは，年末からリッチにタイへ海外旅行に行く計画をしています。

ゆたか：こいつは，めでタイ……ごめん。出だしから，こけてしまった。

うさ子：いいのよ。気を取り直して，お手紙を読みましょうね。

Tさん(三重)：前回の連載を読んで，すこし疑問に思ったのですが，宇宙人森山さんの「宇宙人」とは何のことでしょうか？　ほかのスタッフの方にも，変な肩書き(？)がついてますが，これはいったい何なのですか？

うさ子：変な肩書きは，いわゆる通称，愛称です。たとえば，アマチュアCGA界には森山さんが３人もいらっしゃるので，単に「森山さんからお電話がありました」では誰なのかわからないわけです。

ゆたか：宇宙人森山さんの「宇宙人」は，初期の作品「超強力宇宙人」に由来しています。この作品がCGAコンテストに入賞したとき，月刊ASCIIに森山さんが写真入りで掲載されたのですが，その写真の下のコメントで，本来なら“「超強力宇宙人」の森山氏”となるべきところが，“超強力宇宙人の森山氏”となっていました。それを見たスタッフが「そうか，森山さんって宇宙人だったのか……」といったのがきっかけです。

Oさん(東京)：CGAマガジン第４号が10月18日に届くと書いてあったのですが，まだ着きません。遅れてるんスか？　それとも，やっぱ大阪は，しょっちゅうゴモラとか出て忙しいスか？

うさ子：遅れてすいません。でも，この雑誌が発行されるころには，とっくに届いていると思います。定期購読者で，まだ着いていない方がいらっしゃいましたら，ご連絡ください。

ゆたか：大阪でも淡路は，比較的怪獣が通らない場所です。怪獣はだいたい大阪城を目指しますから。それより心配なのは，こんどのCGAコンテスト会場です。なにしろ，国会議事堂と最高裁裁判所と皇居に囲まれていますから。「どれを壊そうかな」と迷っているときに，「プチッ」とつぶされそう。３月の上旬ってのは，よく怪獣が上陸する季節なんでしょうか？　東京在住の方，教えてください。

マツヤデンキ(淡路)：東淀川区淡路５－17－２　102号　松井プロジェクトチーム様　冬のボーナス特別祭のおしらせ(以下省略)

ゆたか：プロジェクトチームDōGA松井様の間違いです。不気味だから，訂正してください。

ASAHIパソコン編集部(東京)：「Paso」創刊記念パーティーへのお誘い(以下省略)

ゆたか：「出席します　欠席します」　う～ん。すいません。

うさ子：あのぉ，ゆたかさん。それもＱ＆Ａに来たお手紙じゃないと思うんですけど……。

Hさん：画像データを，イエロー，マゼンタ，ブルーに分解して，「プリントゴッコ」のフルカラー印刷という方法が使えるようなツールを作っていただきたいのですが。そうすれば，CGAシステムで作成した画像を，安価で大量に印刷することができるようになります。なにとぞ，ご考慮のほど，お願い申し上げます。

うさ子：いいですね。うさ子も欲しいな。

ゆたか：そうですね。前向きに検討しましょうか。しかし，次のCGAマガジンで配布するとしても，今回の年賀状には間に合いませんね。ごめん。

Hさん：次号のCGAマガジンは，「飛ぶもの」特集だそうですが，羽根が生えて，飛んでゆくお札なんてどうですか？

ゆたか：簡単に作れるんとちゃうかな。

うさ子：でも，スキャナで１万円札を読み込んでもいいの？

ゆたか：マッピング画像に使うことが，著作権上問題あるかな。

うさ子：そうじゃなくて，その画像をプリントアウトしたら。

ゆたか：……ちょっと怖いな。やめとこ。

土田さん(芸術祭オープニング)：先月号のOh!X読んだぞ。俺はいつ「DM」の共同制作者になったんだ？

ゆたか：はっはっは。「DM」の企画書とサンプルデータが入ったディスクが届いたときからに決まってるやん。

うさ子：本人の承諾も得ず，いきなり送りつけたんですか？

ゆたか：召集令状みたいなもんやな。赤いディスクに入れて送ったらよかった。

Oさん(東京都)：BETA.XとPATIPIC.Xがコマンドラインからできないのですが，どうしたらよいでしょうか？

うさ子：バグですか？

ゆたか：いやいや，そんなことはないけど，とにかくこれだけの問い合わせでは，意味がぜんぜんわかりません。

うさ子：PESのメニューにないってことでしょうか？　これらのコマンドは初心者向きではないので，PESのメニューには出ません。しかし，ハードディスクにはちゃんと入っていますから，マニュアルを見ながら，コマンドラインから入力してください。

ゆたか：お～，すごい。今回唯一のまともなＱ＆Ａでしたね。

か：すると「Free」の「Pen」で太さを変えるわけやね。

森：それだけじゃなく，極端に太い線を描くときには，やはり「Poly」を使いますよ。それから，もっと細い線も使います。

か：えっ？　私もいちばん細い筆を使っているのですけど……。

森：だから，解像度が512の画像に対してEPA2を使えば，同じいちばん細い筆が，さらに4分の1の太さになるでしょ。そして，アニエフェを加えたあとで512の画像を256に戻してやればいいんですよ。まぁ，256でもできないってことじゃないけど。

か：なるほど。ちょっと待ってください。いま，512で作画し直しますから。

森：いや，それは必要ないですよ。アニエフェを加えるときだけ512であればいいのですから。256の画像をdchange.xで512の画像に引き伸ばせばいいんです。

か：あっ，そうか。それじゃ，512のモードで太い線や細い線が混在している絵を1枚描いてもらえませんか。

森：こんな感じかなぁ(写真M)。

か：おっ，太めの線にまとわりつくように細い線を描くわけですね。

森：そういうことですね。量は細い線のほうが多くなります。「Poly」を使うと，1本の線のなかに，太いところと，細いところが出るのがいいですね。この絵では，中央の「Poly」で描いた稲妻は，視点の近くにあって，あとは遠くにあるってことですね。

か：ちゃんと3次元的なパースを計算しているわけか。

森：それは考えてます。たとえば，写真N-1とN-2を見ればわかりますが，奥のほうの放電と手前の放電では太さが異なるわけです。元絵がパースのついた奥行きのある絵の場合，そのへんはちゃんと計算に入れないと不自然になります。

それから，3つ目のポイントとしては，線の太さや長さに変化をつけるだけではなく，時間的な変化も大切です。

か：というと？

森：前後のフレームで，同じところが同じように光っていてはいけないんです。ピカピカっていう言葉からもわかるように，明るいのと暗いのが交互に点滅するようにするわけです。かまたさんは，アニメートモードを気にして，前のフレームをなぞるように描いているでしょ。確かに，アニメートモードは，前のフレームを確認するためのものですが，同じにするのではなく，前のフレームとは異なるように描かなくちゃ。

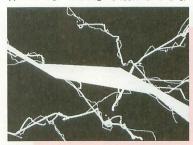
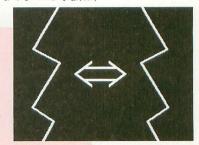
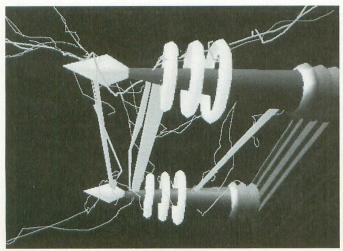
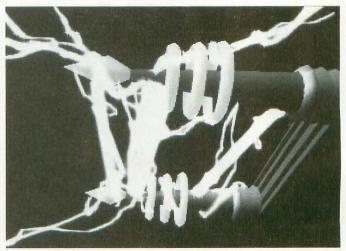
か：あっ，アニメートモードの使い方が，まったく逆なんだ。でも，ちょっと疑問があるんですが。明るいのと暗いのを交互に点滅するようにっていってもいろんな描き方があると思うんですよ。

森：そうですね。

か：たとえば，写真Oのように，ギザギザの曲がり方を左右逆転するとか，あるフレームでは細い線だが次のフレームでは「Poly」で描いたような太い線だとか，もっと極端な話，あるフレームでは右に稲妻が伸びていたかと思えば，次のフレームでは左，次は上というように全然違う場所に描くという方法もあるでしょ。

森：結論からいうと，それらはすべて使います。さらに，もっと大胆な方法もありますよ。あるフレームでは稲妻をたくさん描き込むが，次のフレームではまったく描かないとか。

か：えっ，まったく描かないんですか？

森：ええ。「Next」で飛ばしてしまいます。この手法は，

写真M　森山氏に描いてもらった効果線

写真O　ギザギザを左右逆転する

写真N-1　パースを計算に入れて線を描く

写真N-2　EPA2で光らせる

よく使うというか，ほとんどそうやってますよ。迫力が出るだけでなく，作業量が減るというメリットも大きいですから。

か：そりゃ，何も描かないんだから楽ですわな。そやけど，なかなか大胆やね。

森：大胆というなら，１フレームだけ，全画面真っ白とか，背景の宇宙空間を真っ白にした画像を挿入するっていうこともやります。もっともこれは，過激な爆発とか，視点に近いところの爆発とかの場合ですけど。

か：でも，完全に真っ白の画像はやめてください。RGBがすべて31のような強い信号は，ビデオに録画したときに，信号が乱れてしまいます。

森：ええ，私はRGBを28以下に抑えています。

3Dの原画に工夫する

か：先生，ありがとうございました。変化のつけ方はだいたいわかりました。ほかに注意することは？

森：そうですね。アニエフェは，決してあとからつけ加えるものじゃなくて，最初から計算に入れておくということも大切ですね。

か：構図とかの問題ですか？

森：そうそう。余白がないと加えようがないでしょ。3Dの画像を作る段階で，このカットでは，こんなアニエフェを加えるからこの部分に余白を入れておくとかね。その点，かまたさんがさきほど作ったサンプルの画像は，問題ないですね。無意識のうちにそのあたりがちゃんと計算されているんですよ。

か：へんっ，だてにCGやってませんからね。

森：でも，まだ十分じゃない。

か：すんまへ～ん。

森：実は，アニエフェの一部は，EPA2ではなく3Dでちゃんとレンダリングしてあるんです。たとえば，丸く広がる爆発なんかは，ちゃんとポリゴンの球で作ります。

か：EPA2で円を描いてはいけないんですか？

森：だめってことはないけど，爆発の中心の位置がフレームごとに異なってきたりするし，各パーツが球に飲み込まれる感じが出ない。写真P-1がEPA2による円で，写真P-2がポリゴンによる球。だいぶ違うでしょ。

か：確かにポリゴンのほうが立体感がありますね。

森：私はあまりやらないけど，正確なパースが必要なカットでは，ビームなんかの類もポリゴンで作ります。また，写真Qの飛び散る火花なんかも実はポリゴンで作っていたりします。

か：これは，BOM.Xを使っていますね。

森：使ってません。CADでちゃんと火花の形状を作っています。

か：……何種類ぐらい作るんですか？　それに，どうやって動かすんですか？

森：XY，YZ，ZXの各平面に１枚ずつ，計３枚のイガイガのポリゴンによって構成される形状を１種類だけ用意します(写真R)。この形状を乱数で回転させます。ついでにスケールも適当な幅で，乱数で拡大縮小してやります。飛び散る動きは，フレームソース上で配列を設け，１個１個ちゃんと管理してやります。

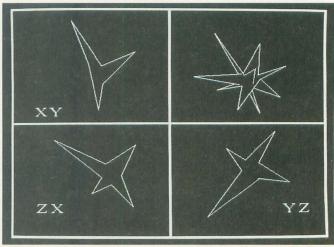
か：なんか，むっちゃ，めんどくさそう。

森：初期値も乱数，飛散方向も乱数ですのでフレームソースが書ける人には案外楽にできると思いますが。サンプルのリスト１を参考にしてください。関数の形になっています。

か：まぁ，この火花のテクニックは，ちょっと上級者向

写真P-1　EPA2による円

写真P-2　ポリゴンによる球

写真Q　ポリゴンで作成した，飛び散る火花

写真R　XY，YZ，ZXの各平面に１枚ずつ形状を用意する

けって感じですね。ほかに工夫はありますか。
森：光源関係があります。レンダリングに時間がかかるのですが，点光源も多用します。
か：まさか，各フレームの稲妻の位置をあらかじめ考えておいて，そこに点光源を配置するとか？
森：そこまではしませんが，点光源を乱数で与えておいて，作画させた画像の一部に極端に明るいところができれば，その近くに大きな稲妻を描きます。また，火花などは，火花自体が乱数なのですから，点光源も乱数で配置すればよいのです。
か：そうですよね。いちいち位置関係を考えていたらたいへんですよね。

CGAマガジン編集部より

よくやった！　との声も大きいCGAマガジン第４号だが，例によって，バグの情報が入ってきている。すまん。あれだけの内容をすべてのマシンのすべての環境で十分チェックするのは不可能に等しいのだ。
許してくれ〜。

*　　　　*

担当者より：今号はたくさんバグを出してしまいました。申し訳ありません。作画を実行してうまくいかなかった場合は，以下のようにバッチファイルやフレームソースを修正して実行し直してください。
なお，「RENDXVI」となっているところは，機種によって「REND　」または「REND030」になっていることがあります。

●Scene　#02のCut05からCut15までを作画したあとにScene #02(64×64)の作画を実行した場合，途中で，「コマンドまたはファイル名が違います」というメッセージが出て，異常終了してしまうバグの修正方法。

S02.BAT　　311行目
（誤）CONV256TO64　C256¥C15_001 C64¥C15_001
（正）PIC64　　　　C256¥C15_001 C64¥C15_001

●メインメモリ2Mバイトのマシンで，Scene　#03のCut16からCut20までを作画したあとにScene #03(64×64)の作画を実行した場合，画像が生成されずに異常終了してしまうバグの修正方法。

S03_2M.BAT　23行目（以下 49，74，119，153，178行目も同様）
:c16_conv
cd C16
（追加）md C64

●メインメモリ2Mバイトのマシンで，Scene　#06のCut34からCut37までを作画させたのち，Scene #06 (64×64)の作画&アニメーションを実行した場合，アニメーション時のC35A_071.PICからC35A_112.PICにかけて，画像が黒い墨を撒いたようになってしまう(CRDがかかっていない)バグの修正方法。

S06_2M.BAT　72行目
（誤）CRD C64¥C35A_001 /OC64¥C35A_001 /N70
（正）CRD C64¥C35A_001 /OC64¥C35A_001

●Cut36で「END」の文字が飛んでくるところで，青色の飛行機が画面右からフレームインしてきてしまうバグの修正方法。

・4Mバイト以上の場合
C35A_.FSC　　55行目
（誤）｛ mov (¥div(TMX)¥ ¥TMY¥ 0) target ｝ rotx(¥BRX¥) obj BLCF BLL
（正）｛ mov (¥div(TMX)¥ ¥TMY¥ 0) target ｝ rotx(¥BRX¥) (:obj BLCF BLL:)
C35A_.FSC　　57行目
（誤）(:#if (fno<151) obj BLCF BLL #else #endif:)
（正）#if (fno<151) obj BLCF BLL #else #endif
・2Mバイトの場合
C35AX.FSC　　55行目
（誤）｛ mov (¥div(TMX)¥ ¥TMY¥ 0) target ｝ rotx(¥BRX¥) (:obj BLCF:) BLL
（正）｛ mov (¥div(TMX)¥ ¥TMY¥ 0) target ｝ rotx(¥BRX¥) (:obj BLCF BLL:)
C35AX.FSC　　57行目
（誤）(:#if (fno<151) obj BLCF BLL #else #endif:)
（正）#if (fno<151) (:obj BLCF:) BLL #else #endif
C35AY.FSC　　55行目
（誤）｛ mov (¥div(TMX)¥ ¥TMY¥ 0) target ｝ rotx(¥BRX¥) obj BLCF BLL
（正）｛ mov (¥div(TMX)¥ ¥TMY¥ 0) target ｝ rotx(¥BRX¥) (:obj BLCF BLL:)
C35AY.FSC　　57行目
（誤）(:#if (fno<151) obj BLCF BLL #else #endif:)
（正）#if (fno<151) obj BLCF BLL #else #endif

●Cut37を作画してアニメーションさせたのち，2度目にCut37を実行したとき，正しくはアニメーションが始まるべきところが，再び作画を始めてしまうバグの修正方法。

C35D.BAT　　3行目
（誤）if exist C35F¥C256¥C35F_030.PIC goto anim
（正）if exist C35E¥C256¥C35F_030.PIC goto anim

●メインメモリ4Mバイト以上のマシンで，Scene #06(64×64)の作画&アニメーションを実行した場合に，C35A_001.PICからC35A_150.PICで背景のX68000が小さく作画されてしまうバグの修正方法。

S06.BAT　　43行目から55行目まで
（誤）del C35A_.FRM
FF /R /s1:150 C35AX.FSC
if errorlevel 1 goto err
RENDXVI　/A2 /G　/C64x64 /P96,96 "/EIDEL $B" /HC64¥S_001 /OC64¥H_001　C35AX.FRM　@C35AX.IND
…
del C35AY.FRM
（正）RENDXVI　/A2 /G　/C64x64 /P96,96 "/EIDEL $B" /HC64¥S_001 /OC64¥C35A_001　C35A_.FRM　@C35A.IND
if errorlevel 1 goto err
del C35A_.FRM

●Scene #06(64×64)の作画をしアニメーションを実行した場合に，「THE END」の背景に星がないバグの修正方法。

・ハードディスクに展開する場合
S06.BAT　　187行目（4Mバイト以上の場合）
S06_2M.BAT　267行目（2Mバイトの場合）
（誤）RENDXVI　/A2 /G /S20:20 /OC64¥H_020　C35D_.FRM　@C35D.IND
（正）STAR /H5000 /S20:20 /OC64¥S_ C35D_.FRM
RENDXVI　/A2 /G /S20:20 "/EIDEL $B" /HC64¥S_020 /OC64¥H020　C35D_.FRM　@C35D.IND
・フロッピーディスクに展開する場合
FS06.BAT　　178行目（4Mバイト以上の場合）
FS06_2M.BAT 433行目（2Mバイトの場合）
（誤）RENDXVI　/A2 /G /S20:20 /OB:¥C35D¥C64¥H_020　C35D_.FRM @C35D.IND
（正）STAR /H5000 /S20:20 /OB:¥C35D¥C64¥S_ C35D_.FRM
RENDXVI　/A2 /G /S20:20 "/EIDEL $B" /HB:¥C35D¥C64¥S_020 /OB:¥C35D¥C64¥H_020　C35D_.FRM　@C35D.IND

森：でも，たとえばビームの発射シーンや，エンジン点火シーンなどは，配置すべき位置がはっきりしているので，ちゃんと計算しておきますよ。

それから，テクニックのひとつとしてぜひ知っておいてほしいのですが，極端に明るいカットでは，光源を暗くするという方法が有効です。

か：はっ？　逆でしょ。明るいカットでは，明るくしなくちゃ。

森：そう思うでしょ。ところがちがうんですよ。写真S-1とS-2を比べて見てください。ビーム光線は，どちらが明るく見えますか？

か：あっ，なるほど。確かに，S-1のほうが明るいっていうか，全力をぶつけたビームって感じがしますね。これは，ビーム自体の明るさは同じだけど，S-1のほうは，ほかの部分が暗いために，ビームの明るさが強調されるわけですね。

森：まぁ，そういうこと。ほかの部分は，単に暗くするというよりアンビエントの値を下げて陰影をつけるって感じかな。それと，もうひとつ意味がある。さっきも解説したように，緊張感を高めるためには，ピカピカと明るいのと暗いのを点滅させるというのがポイントです。だから，写真Tのように，明るい光源のカットのなかに光源が暗いフレームが挿入されると，画面全体が点滅するようになって，明るさが強調されるわけです。

か：へぇ～，こうしてみると，いろんな隠し技があるんですね。

森：苦労してます。あと，爆発球やビーム，火花のようにアニエフェの一部を3Dでレンダリングするときは，512の解像度で作画することがあります。

か：あぁ，それはわかります。256でアンチエイリアスをかけると，ぼけた部分が光らせる色と異なってくるから

リスト1

```
(:      破片制御関数     :)
#define MAX_HAHEN        200
#init   ¥ ha_x[ MAX_HAHEN ],
                                ha_y[ MAX_HAHEN ], ha_z[ MAX_HAHEN ] ¥
#init   ¥ ha_vx[ MAX_HAHEN ],
                                ha_vy[ MAX_HAHEN ], ha_vz[ MAX_HAHEN ] ¥
#define HA_ARGS ha_x[], ha_y[], ha_z[], ha_vx[], ha_vy[], ha_vz[]

(:      破片配列を初期化する     :)
#func Init_Hahen( HA_ARGS )
        #rep ( i, 0, ( MAX_HAHEN - 1 ) )
                #do ¥ ha_x[ i ] =  0 ¥
                #do ¥ ha_y[ i ] =  0 ¥
                #do ¥ ha_z[ i ] =  0 ¥
                #do ¥ ha_vx[ i ] = ( rand() * 512 ) - 256  ¥
                #do ¥ ha_vy[ i ] = ( rand() * 512 ) - 256  ¥
                #do ¥ ha_vz[ i ] = ( rand() * 512 ) - 256  ¥
        #endrep
#endfunc()

(:      破片を動かす     :)
#func Move_Hahen( HA_ARGS )
{
        #rep( i, 0, ( MAX_HAHEN - 1 ) )
                #do ¥ ha_x[ i ] = ( ( ha_x[ i ] ) + ha_vx[ i ] ) ¥
                #do ¥ ha_y[ i ] = ( ( ha_y[ i ] ) + ha_vy[ i ] ) ¥
                #do ¥ ha_z[ i ] = ( ( ha_z[ i ] ) + ha_vz[ i ] ) ¥
                {
                        mov( ¥ ha_x[ i ] ¥ ¥ ha_y[ i ] ¥ ¥ ha_z[ i ] ¥ )
                        scal( ¥( rand() * 0.2 ) + 0.01¥
                              ¥( rand() * 0.2 ) + 0.01¥
                              ¥( rand() * 0.2 ) + 0.01¥ )
                        rotx( ¥rand() * 360¥ )
                        roty( ¥rand() * 360¥ )
                        rotz( ¥rand() * 360¥ )
                        obj Hahen
                }
        #endrep
}
#endfunc()
```

DōGA法人化への道

相変わらず，刻々と状況が変化する法人化問題だが，なんとかまとまりつつある。税理士さんとよく相談してみると，我々の意思がどうこういう以前の問題で，法律的に選択の余地はほとんどないらしい。

まず，アマチュアのプロジェクトチームDōGAは，そのまま任意団体という形で存続させる。任意団体というのは，以前紹介した「人格無き社団等」に相当するもので，大学のクラブや町内会など，特に営利を目的としない団体で，公益法人としての認可は得られていない団体だ。ひとことでいえば「アマチュア」だが，たとえば「くもん式」で有名な「くもんの会」もどうやら任意団体らしく，全国規模の組織にもなり得るらしい。

なぜ，任意団体にするのかというと，答えは簡単で，公益法人にするにしろ，協同組合にするにしろ，まずはいったん任意団体として活動したうえで，その活動実績がないと認可が得られないからだ。

任意団体は，法的には公益法人に準じた扱いを受ける。基本的に非営利団体だし，また，将来本当に公益法人を目指すためにも収益活動はいっさい行うべきではない。少しでも収益活動を行うと，税制上の問題も発生する。

しかし，そうすると活動資金はどうするのだろう？　任意団体の資金源として認められそうなものを調べると，会費とカンパがある。ただ，現在DōGAでは，会費はゼロなので，今後もやっぱりカンパが唯一の収入源となる。はっはっは，ちょっと不安かな？　でも，いままでちゃんとカンパだけで成り立っていたんだから，今後も努力して，ユーザーが思わずカンパしてしまうようなよいものをどんどん出していこう。皆さんもご協力お願いします。

だが，ここで大きな問題が出てきた。いっさいの収益事業が認められないのであるなら，CGAマガジンはもちろん，マニュアルやコンテストのビデオ配布ができなくなる。確かに，あれは実費なんだから，収益事業には相当しないといいたいところだが，そんな理屈はお役所には通じないそうだ。こいつは困った。

そこで出てくるのが株式会社ドーガ。これらの配布物は，アマチュアのDōGAが企画，開発などを行い，(株)ドーガが販売するという形式をとる。(株)ドーガは，株式会社なんだから収益事業をしても，なんら問題はない。収入に応じて，ちゃんと税金を納めればよいのだ(とすると，次回のCGAコンテストビデオには，消費税がかかるぞ……いやだなぁ)。

ただ，ここで疑惑がひとつ出てくる。

「(株)ドーガは，アマチュアが作ったものを集めて，丸儲けしてるのではないか？」

しかし，ご安心を！　(株)ドーガの株は100%アマチュアのDōGAが持っているのだ。アマチュアのDōGAは，配布物を可能な限りの低価格で多くの人に配ることを目的に(株)ドーガに販売をさせているのだから，定価を上げて暴利をむさぼるようなことは，株主権限として断固許さない。(株)ドーガの会計報告はもちろん，無駄な出費がないかどうか，常に監視を行うわけだ。

だから，今度のCGAコンテストのビデオも，正当な理由(120分になったとか)もないのに，前回と比べて定価が上がるなんてことは，アマチュアのDōGAが許さないのだ。そして，ビデオを申し込むとき，ついでにカンパしてくれた分については，ちゃんとアマチュアのDōGAに渡るという仕組みになるわけだ。

もっとも，(株)ドーガが毎年，赤字になって倒産してもらっては困る。そのへんの手綱さばきが難しい。「生かさず，殺さず」ってやつだな。

ということで，プロジェクトチームDōGAは，名実ともに非営利団体となるわけだが，そのためにはまず，任意団体にならなければいけない。ではその任意団体の条件は何か？　いちばん難しいのが正会員の決め方だが，そのあたりの話はまた次回に！

でしょ。

森：そのとおり。512のアンチエイリアスなしで作画しておいて，アニエフェを加えたあとで，アンチエイリアスをかけながら256に変換する。

そういえば，かまたさん，先月号のなかで，256に変換する方法として，「Paste」の「Differential」を使うって紹介していたでしょ。

写真S－1　光源を暗くする

写真S－2　光源を明るくする

でも，あの方法では，アンチエイリアスがかからないし，256になった画像をセーブできないから，現実的には無理ですよ。「Sys」の「Screen Mode」ならアンチエイリアスつき解像度変換をするし，マクロ定義の対象にもなるから使うならこっちでしょう。

か：あっ，そうか。ごめんなさい。

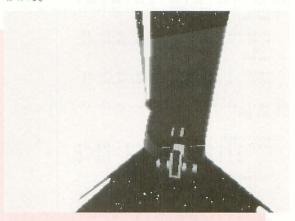
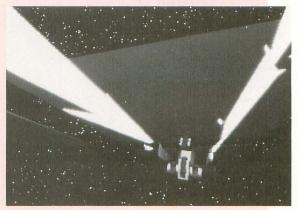
森：でも，なんだかんだいっても512の画像ファイルは扱いにくいですし，アンチがかかった画像もそれなりに光るので，256でやってしまうことのほうが多いかな。

か：なるほど。そのへんに注意して，あとは実際にやってみるしかなさそうですね。

森：だいたいアニエフェで，リアリティなんか追求するもんじゃないですよ。むしろ誇張された表現のほうが，まだほかとの調和を図りやすい。特に，1コマ1コマ絵になっているかどうかなんて気にし始めたら，まず確実にこけます。ダサいカットができます。

3Dの原画の修正

か：それでは，先ほどのサンプルカットに実際に森山さんの手でアニエフェを加えていただきましょう。まずは3Dの原画を修正してみます。

最初に，爆発球を加えましょう。TAMEN.Xで球を作って，/Cオプション(面の細かさ)は2で十分だよな。アトリビュートはアンビエントを1.0にして，色はRGB＝0.25，0.50，0.75ぐらいかな。どの位置にどのくらいの大きさで置くのかは，FFE.Xでだいたいのアタリをつけておく。これをもとに先ほどのフレームソースを改造するわけだ。

写真T　暗いカットを挿入して画面全体を点滅させることで，明るさを強調する

11フレーム目でビームが貫通して，12フレーム目から爆発球が発生するわけやけど……。森山先生，質問なんですけど，どんな割合で大きくするんですか？

森：「先生」はやめて。大きくするのは適当でいいんとちゃいますか。いろんな爆発があるでしょうし。私は，フレームソースに独自の関数を読み込んでいますけど。イメージからすると，初めは急速に大きくなって，だんだんゆっくりになる……現実から考えるとおかしいね。

か：なぜ，おかしいの？

森：だって，宇宙空間なんだから，機体から飛び出したものは，一定速度で離れていく。つまり，爆発球の半径は，フレームナンバーに比例するだけというのが正しいんとちゃうかな。

か：そうやけど，毎秒一定量のガスが噴出していて，それが輝いていると考えると，球の半径は，最初は急激に大きくなり，球の体積が増すにつれてだんだんゆっくりと大きくなるわな。

森：えっと，その場合どんな式になるんだろ。

か：体積は，半径の3乗に比例するんだろ。そして，11フレーム目の半径は0で，20フレーム目ぐらいで300になってほしい。爆発球の形状(MARU.SUF)の半径は100だから……，うっ，この歳になって，数学やるのか？　3乗根は，1/3乗と同じ，FF.Xには，XのN乗の関数，pow(X,N)があるから……こんな感じかな(リスト2)。

ついでに，20フレーム目と23フレーム目に，平方根に比例した爆発球を，そして27フレーム目からは単純に正比例した爆発球を置いてみました。

森：ボコボコボコっと，連続して爆発していくわけですね。レンダリングしてみましょう。……おっ，なかなかいい感じ出てるじゃないですか。

か：平方根と3乗根との差はあまりないって気もするけど，これはなかなか使えますよ。爆発球ができたら，火花は面倒だから今回はパスして，あとは光源関係を修正しなくちゃ。

森：それは無意味とちゃうかな。

か：えっ，なんで。

森：だって，このサンプルでは，宇宙戦闘機以外の物体が何もないやんか。あたりまえやけど，いくら光源を工夫しても，照らされるものが少ないと効果ないですよ。

か：そうか，それで写真S-1にも，差がわかるように意味のないモノリスを並べてたのか。

森：背景があるとか，まわりにほかの宇宙船があるとか，あるいは宇宙船のアップになっているとかのときに，光源を工夫すべきやね。

か：じゃあ，とりあえず原画は，このままでいきましょう。この256の画像を，512に変換して，はい，準備できました。それでは先生，お願いします。

森：だから，「先生」はやめて。

リスト2

```
#frame( fno, 1, 30 )
@5.3@
fram{
        light pal( rgb ( 1.00 1.00 1.00 ) 3.00 2.00 2.00 )
        { mov   ( ¥div( 100, -400, -650,  1, 20, 30,fno )¥ -900  0
                ) eye deg( 40 )
        }
        { mov   ( ¥div( 400, 800, 1200,  1, 20, 30,fno )¥
                  ¥div(   0,2900, 4300,  1, 20, 30,fno )¥
                  -100
                ) target
        }
        { mov   ( ¥div(   0, 800, 1200,  1, 20, 30,fno )¥
                  ¥div(   0,2800, 4200,  1, 20, 30,fno )¥
                  0
                )
          rotx  ( ¥div(  45, -10,  -30,  1, 20, 30,fno )¥
                ) obj ohx

#if( fno > 11 )
                { mov ( -220  -320 0 )
                  scal( ¥ 3 * pow( ( fno-11 )/10 , 1/3 ) ¥
                        ¥ 3 * pow( ( fno-11 )/10 , 1/3 ) ¥
                        ¥ 3 * pow( ( fno-11 )/10 , 1/3 ) ¥
                      )
                  obj maru
                }
#endif
#if( fno > 19 )
                { mov ( -380  20  80 )
                  scal( ¥ 2 * sqrt( fno-19 ) ¥
                        ¥ 2 * sqrt( fno-19 ) ¥
                        ¥ 2 * sqrt( fno-19 ) ¥
                      )
                  obj maru
                }
#endif
#if( fno > 22 )
                { mov ( 580 480 -80 )
                  scal( ¥ 3 * sqrt( fno-22 ) ¥
                        ¥ 3 * sqrt( fno-22 ) ¥
                        ¥ 3 * sqrt( fno-22 ) ¥
                      )
                  obj maru
                }
#endif
#if( fno > 26 )
                { mov ( 100   0   0 )
                  scal( ¥ 3 * ( fno-26 ) ¥
                        ¥ 3 * ( fno-26 ) ¥
                        ¥ 3 * ( fno-26 ) ¥
                      )
                  obj maru
                }
#endif
        }
}
#endframe
```

おわりに

あっ，いきなり「おわりに」に入っちゃいましたね。盛り上がってきたところなのに，残念ですね。ページ配分を間違えたのが，失敗でした……ごめんなさい。

実は，アニエフェのノウハウはまだまだあるんです。さらに，応用編まであるんです。ということで，次回もこの「EPA2補講」を続けさせていただきます。

それまでにぜひ一度，皆さん自身の手で挑戦してみてください。というのは，原稿を書いていて思ったのですが，単に人の話を聞くより，その前に自分で苦労して，失敗してから聞くほうがよくわかるし，身につくのです。

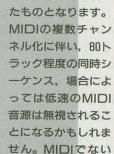
今回の原稿を読んで，「アニエフェのコツがなんとなくわかった」とか「けっこう簡単そう」と思った方も多いと思います。しかし，実際に描いてみると，はっきりいって，今回のノウハウだけでは，かっこいいアニエフェを加えることは難しいということがわかるはずです。何が難しいのか，どうしてうまくできないのか，それを体験してから，次号を読んでいただければ幸いです。

それから，関係ないけど，寺島先生，目がハートは恥ずかしいからやめてくださいね。

[特集]

Z-MUSICシステム ver. 2.0

お待たせしました。ついに新しいZ-MUSICシステムの完成です。すでに入手して使っている人もいるかもしれませんね。
ドライバ自体の，Oh!Xによる正式リリース版はver.1.01，ver.1.10，ver.1.52の３種類ですが，事実上，ver.1.10相当の機能までしか解禁されていませんでした。
Z-MUSICは「システム」であって，ドライバ本体（ZMUSIC.X）のみならず，ドラムサウンドライブラリ，サポートプログラムまでをすべて含んで，ひとつの世界を形成しています。それがようやくすべてまとまったかたちになりました。
今回のバージョンアップで,Z-MUSICシステムもひととおりのものが完成した感じになります。新しい機能を存分に使用してみてください。

もちろんこの瞬間からver.3.0への模索も始まっています。このver.3.0は当初の構想にあったver.2.0に相当します。音楽演奏のためだけのドライバとして，可能性を追求したものとなります。MIDIの複数チャンネル化に伴い，80トラック程度の同時シーケンス，場合によっては低速のMIDI音源は無視されることになるかもしれません。MIDIでない外部音源を制御するかもしれません。それはこれまでのZ-MUSICシステムとは別の世界です。ですからver.3.0がリリースされてもver.2.Xがなくなるわけではありません。Z-MUSICはより広く音楽に対応していくことでしょう。

CONTENTS

Z-MUSIC

[特集]
Z-MUSICシステム
ver. 2.0

新機能の概要

バージョン2.0ができるまで

Nishikawa Zenji 西川 善司

初版以来，苦節2年。ついに完成した新しいZ-MUSICシステム。ここではZ-MUSIC ver.1.1からバージョンアップされた点を中心に解説していきます。

V1からV2へ

「Z-MUSIC システム ver.2.0」ついに発売にこぎつけることができました。初版である「ver.1.0」がリリースされてからなんと2年が経ってしまいましたが，なにも菓子をツマミながらお茶を飲んで隠居していたのでもなく，真の格闘家になるために修行に出ていたのでもなく，ずっとバージョンアップに励んでいたのです。

初版発売後「Z-MUSICを発展させるためには多くの人に使ってもらわなくてはならない」と考え，多くのX68000ユーザーとの意見交換ができるパソコン通信を始めることにしました。そこで最寄りのネット(PC-VAN X1CLUB,MIYA-NETなど)でZ-MUSICのサポートSIGを作ってもらい，ここからユーザーの声を汲み取ることを始めました。予想以上に反響があり，その結果，さまざまなユーザーから新機能の追加要請やバグ報告が寄せられました。こうして機能強化，バグ撲滅を繰り返していくうちに月日は流れていったのです。

もともと，パソコン通信界ではさまざまなミュージックツールが出回っていたようで，そこにポッと出のZ-MUSICが簡単に受け入れられるはずもなかったのですが，効果音モード，ライセンス放棄(商的利用の許可)などが受け入れられたようで，いまでは同人ソフトや市販ゲームにまで組み込まれたりすることもあり，作者としては大変喜ばしい状況となりました。

また，優秀なサポートツール，周辺ツールが続々とユーザー側の手によって生み出され，環境が整えられていったのも利用者増加の一因です。これらのツールはいずれ「Z-MUSICサポートツール集」としてパソコン通信利用者以外の皆さんの手元へ届けられることでしょう。

それでは，ver.1.1(以下V1)から強化された機能の話題を中心にver.2.0(以下V2)を紹介していきたいと思います。

モジュレーション機能の強化

まず，V1のFM音源パート部分では，モジュレーション波形が三角波のみでしたが矩形波，ノコギリ波(鋸歯波)，ノコギリ波シングルの3種類が追加され，合計4種類の波形から選択できるようになりました。

また，V1では実現できなかった音量モジュレーション(トレモロ)も実現できるようになり，さらにこれを発展させたワウワウ機能も搭載されました。ワウワウ機能とは任意のオペレータのアウトプットレベルに対してモジュレーション処理を行うことにより音色を周期的に変化させるものです。

プリセット波形のほかにユーザーが作成した波形を登録し，これをモジュレーション波形として使用できるようになりました(波形メモリ機能)。音程/音量をリアルタイムにユーザー波形の下で変化させることができるため複雑な演奏表現や効果音の作成に有用です。

MIDIパートではV1では，コントロールチェンジのビブラート機能によるピッチモジュレーションしかなかったのですが，V2からはFM音源と同様に4種類のプリセット波形やユーザー波形を用いてモジュレーションが行えるようになりました。つまり，シンセサイザのベンダーホイールをZ-MUSICが自動的にかつ周期的に回してくれるのです。これまでとはまったく違った新しい演奏表現が開拓できるでしょう。

もちろん，これらの波形を任意のコントロールチェンジに対しても出力する(ARCC機能)ことができます。たとえばコントロール7番(または11番)に対して行えばFM音源パートとまったく同じように音量モジュレーション(トレモロ)も実現できるわけです。

MIDIパートの柔軟性強化

V1FM音源とMIDIではその音程管理の構造上の違いから，持続音の音程変更に関して表記が同じでも効果が異なるということがありました。

たとえば，スラーを表現する場合，

C&C#

としてもV1では，FM音源部ではCの持続音がC#へ上がりますが，MIDIではこの表記ではC,C#というふうに2つの音が発音されるだけです。FM音源部と同様の効果を実現するためにはMIDIでは，

C& @B683 C

とピッチベンダーを駆使しなければならなかったのでした。ポルタメント後の持続音に関しても同様の相違点が見られました。

そこでこういった相違点を解決するモードが搭載されました。これでMIDIパートもFM音源パートのような感覚でグリッサンドやスラーが実現できます。

変化記号の強化

#や♭が多い調(嬰ヘ長調，変ホ短調など)の楽譜は入力の際1音ずつ変化記号を表記していたのでは見づらくなりますし，入力ミスを犯しやすくなります。そこで調号をトラックごとに設定することができるようにしました。

これは[K.SIGN]という新しい命令で行われます。

[K.SIGN F+]

とすると，Fに#がついたということで，ト長調の指定となります。

PCM8.Xへの正式対応

PCM8.Xとは江藤啓氏の制作したADPCMドライバです。本来ならば単音しか発

声することができないX68000のAD PCMをソフトウェア的な処理だけで8声まで多重再生できるように拡張してくれる画期的なアプリケーションです(今回発売されたV2にも収録されています)。

実はV1から隠し機能で対応していたのですが，これに正式対応し，PCM8.Xと併用時にはZ-MUSIC上でPCMを8和音まで発声することができるようになりました。またAD PCMパートの音量も可変となり，いままでとは違ったAD PCM音源の利用法が切り開かれたといっていいでしょう。

実際，今月号の特集に掲載されている舘野氏の曲などはストリングスパートなどをAD PCMで演奏させ，内蔵音源のみとは思えないアンサンブルを実現しています。

もちろん効果音モードにもPCM8.Xを利用することができます。そのときにはAD PCM音源8声のうち音楽演奏に6声，効果音に2声というような割り当てが可能です(ただし，PCM8.Xは演算量が多大なためXVI以上のCPUパワーがないと，著しくマシン全体の処理スピードが落ちる場合があります)。

また，PCM8.Xでは16ビットPCMや8ビットPCMの再生機能も備わっておりましたが，これにも正式対応しました。

PCMバンク機能

AD PCMを使用する際には仮想の鍵盤1つひとつに音を割り当てていきました。PCM8.Xの登場で新しいAD PCM音源の可能性が切り開かれ，音程を持つ一般的な楽器をこの仮想鍵盤に割り当てていく機会が多くなりました。この仮想鍵盤が不足する事態が起こってきたのです。そこでV2ではこの仮想鍵盤を最大4つまで持つことができ，音色切り替えコマンドで任意に切り替えることができるようになりました。

こうして，FM音源部やMIDIパートのようにひとつのトラックで複数のPCMセットを切り換えて使用することができたり，または音のカテゴリにあわせて管理したりすることができ，いっそうAD PCMパートがシーケンスしやすくなりました。

新PCM音追加収録

もはや標準ともなりつつあるV1のライブラリは1音も漏らさず収録されています。これに加えて今回はさらにクオリティを極めるため，専用サンプラーでサンプリング，これをX680x0へ移植という方式で新たなAD PCM/PCMライブラリを構築しました。

PCM8.Xの登場により，AD PCMにリズム/パーカッションだけでなく一般的な楽器音までも発音させて曲を作成する可能性が見出されました。

そこでV2では，V1では数音しか収録しなかった一般的な楽器音を数多く収録しました。ベース，ギター，ストリングス，ブラス，ピアノなどはユーザーの希望する音長に再構成できるように音のアタック部分と持続音部分と分けて収録しています(詳しくは後述)。

また，V1発売後比較的要望の多かった効果音なども数多く収録しました。爆発音や兵器の発射音をはじめ，波の音やヘリコプターの飛来音などゲームの効果音や前衛音楽を制作するときに役立つでしょう。

このほかV1ではいま一歩クオリティに不満だった一部のリズム楽器や最近の流行のリズム音なども収録してあります。

おそらく，アイデアと根気次第で，PCM8.Xを駆使すればAD PCM音だけの曲を作ることもウソでなく可能です（X68000XVI以上なら，かな？）。

新周辺ツールZPLK.R

今回ディスクに追加収録されたPCM音のいくつかはAD PCM形式だけではなくて16ビットのリニアPCMのものもあります。

こちらは音のアタック部分と持続音と2つのファイルに分けて収録されたものがあります。たとえばブラスの音を「プワー」と奏でたときに，いつまでも音を伸ばしたいときには「プワー」のうち「ワー」をいつまでも奏でることで実現できます。

つまりアタック部分とは「プワー」の「プ」，ループ部分とは「ワー」で，これらを2つのファイルに分けてあるのです。「プ」にこの「ワー」の部分を任意の数だけ付け加えれば，任意の長さの楽器音を作ることができるのです。このときの演算処理の誤差がAD PCMと16ビットPCM方式では圧倒的に後者のほうが少ないため，このデータ形式が採用されました。

ここで16ビットPCMファイルを活用するために生まれたのが「ZPLK.R」です。16ビットPCMだけではなくAD PCMとのリンクやその逆も可能です。

また，非常に困難といわれていたピッチベンドやHz単位の周波数変換などもサポートされ，さらにエンベロープの変更も可能で，AD PCM/PCMファイルの編集加工ツールとしての機能のほかに音色の編集機能も備えています(「ZPLK.R」の機能はすべて，V1からお馴染みのZVT.Xに新規搭載されているため，視覚的な編集はこちらで行うと便利です)。

これからのOh!X LIVEには「ZPLK.R」の処理内容を記したBATファイル付きの演奏データが続々登場してくるでしょう。

畳み込み機能搭載

「ZPLK.R」と「ZVT.X」に「畳み込み機能」が搭載されました。これは「音の結婚」ともいわれる演算処理で，2つの音色の特徴が合成され，まったく別の音へ生まれ変わらせることができます。畳み込みはSF映画の効果音作り(怪獣の鳴き声やレーザー砲の発射音など)の際にも活躍していますし，デジタルエフェクタの残響効果もこの処理で作られています。

たとえばディストーションギターと人間の声を畳み込むとロボットの声のような無機質な音声へ生まれ変わります。

現在，整数演算による基本的な方法で処理しているため結果が出るまでに非常に時間がかかってしまうのが欠点ですが，まったく新しい音を生み出す可能性を持ったこの機能，一度体験されてみることをオススメします。

詳しくは今月の特集記事66ページをご覧ください。

アプリケーションをCやBASICで

X-BASIC用外部関数「MUSICZ.FNC」には，新機能を駆使するための新コマンド(波形メモリ登録コマンドなど)のほか，BASICでの周辺ツール開発を手助けするための関数が追加されています(たとえば，AD PCM←→16ビットPCMの変換関数など)。Z-MUSICに対応したPCMエディタや音色エディタ，MIDIデータファイリングツールなどいままでアセンブラでないと困難だった開発が手軽に行えるようになりました。

MUSICZ.FNCを使用したX-BASICプログラムをコンパイルするときや，C言語でZ-MUSICアプリケーションを開発するときに役に立つ，C言語ライブラリも同時に収録されています。周辺ツールでなく，X-BASICやC言語でゲームを開発している場合などにも，これらを用いることで簡単にZ-MUSICに対応させることができる

のです。

3タイプのZ-MUSIC

機能的にはほとんど同じですが，各ユーザーの環境やニーズに合わせて，今回発売されたV2には3タイプのZMUSIC.Xが収録されています。

●ユニバーサルバージョン

昔はX68000といえば68000の10MHz機のみでしたのでソフトもこれで動けば誰からも文句はいわれなかったのですが，時代は進み，現在ではX68000は10MHz機，16MHz機と2タイプ，X68030は25MHzの68030ECが搭載されており，昔のようなことはいっていられなくなりました。さらに改造クロックアップとかアクセラレータとか満開製作所のREDZONE(Compact XVIの24MHz改造機)なども登場し，ひとつのアプリケーションですべてのMPU速度に対応しなければならない状況です。

V1は16MHz機までで完全動作するよう命令レベルで最適化されて作られていましたので当然それ以上の速度のマシンでは動作が保証されなかったのでした。そこでプログラム的に見直されてあらゆるMPU速度に対応するために制作されたのが，ユニバーサルバージョンです。純正MIDI ボードもしくはMIDIなしのユーザーはこのバージョンを使用してください。

●RS-232C-MIDIバージョン

X680x0は拡張I/Oスロット数が少ないため，すでにメモリボードやSCSIボードなど各種拡張ボードでスロットがいっぱいのユーザーはMIDIボードを装着できません。そこで最近PC-9801などでもスタンダードになりつつあるRS-232CでのMIDI出力にZ-MUSICも対応してほしいという要望がV1発売後多数寄せられました。

RS-232CでMIDI出力を行うためには変換アダプタが必要ですが，本体後面のRS-232C端子に装着するだけで拡張I/Oスロットは使用しません。秋葉原などでキットが5000円以下で売られており，一般メーカーの市販品も安価で売られていますので確かに手軽にMIDIが始められます。そういうわけでRS-232C-MIDI版が開発されました。

●POLYPHON対応バージョン

Oh!Xにも広告が掲載されているネオコンピュータシステムから発売中のサブボードPOLYPHONにはMIDI出力端子が用意されています。しかし残念ながら純正MIDIボードとはまったく互換性がありません。これに対応すべく制作されたのがPOLYPHON対応版です。POLYPHON対応版Z-MUSICを使用するには，POLYPHONを購入すると付属してくる「PCM8SB.X」を先に組み込んでおく必要があります。1993年12月現在，PCM8SB.XはMIDI-IN端子がサポートされていませんのでPOLYPHON対応版ZMUSICもMIDI-INがサポートされていません。ご了承ください。

機能縮小版ZMSC.X

標準のZ-MUSICは汎用トラックを80本も持ち，さらにMML言語コンパイラまでも内蔵しています。ゲームに組み込む際などは，演奏データは主にコンパイラをすでに通したZMD形式ですから，コンパイラは不要ですし，一度に演奏可能な最低限のトラック数があれば問題ないことになります。そこで，必要最低限の機能を備えた機能縮小版ZMSC.Xが収録されました。デフォルト常駐量は普通のZMUSIC.Xの半分でたいへん省メモリ/省ディスク指向です。

MMLコンパイル機能やトラック確保，アサインなど一部のファンクションが削除されているためZMD再生専用仕様となっていますが，効果音モードなどはちゃんと備わっているため，いままでどおりゲームなどに組み込んで使用することもできます。

周辺ツールもバージョンアップ

ZP.RやZPCNV.R，ZVT.XなどV1からお馴染みの周辺ツールもバージョンアップしています。機能強化点を簡単に紹介します。

●ZP.R(Z-MUSIC演奏制御ツール)

X68000のキーボードコントロールによる演奏制御機能のバリエーションが増え，また，各ユーザーが任意のキーに任意の制御機能を割り当てることができるようになりました。また，ジュークボックス機能に関しても同様に1曲飛ばしたり，戻したりCD感覚の操作が行えるようになりました。

MIDIケーブルでX68000同士を接続して一度に複数のX68000を同期演奏をさせることができる新機能も搭載されました。

●ZPCNV.R(ZPDジェネレータ)

従来から装備されていたAD PCM音の音程音量変更/ミキシング機能のほかに，フェードイン/フェードアウト，波形の任意の部分のトランケイト(切り出し)機能が搭載されました。ZPLK.Rとあわせて使用することでいままでとは違ったAD PCM音源の使い方を開拓することができるでしょう。

●ZVT.X(サンプリング/編集/加工ツール)

新機能の畳み込み機能のほか，周波数の微調整，ポルタメント，ファイラーの機能強化と，より多機能かつ高機能になりました。最大録音可能時間も10倍以上に拡張されました。

●ZAM.R(演奏モニタリングツール)

堀江孝太郎氏の制作されたMON.Rのバージョンアップ版です。カラフルなキーボード表示，リアルタイムワーク表示機能を備えており，音程，音量，音場などの基本的なパラメータを非常にビジュアルにモニタリングできます。任意のトラックのミュート機能，ZP.Rコンパチキー操作で行える演奏制御で，パートチェックやバランス調整作業も快適に行えZ-MUSICユーザー必携のツールです。

全ソース公開と改訂版マニュアル

V2のプログラムは，ほぼすべてソースリストが収録されています。解析やアプリケーションの制作に役立ててください。

また，マニュアルはV2出版に際して，全書き換えを施しました(これが大変時間がかかったんです)。V1のわかりにくかったところや追加された新機能は図を多く用いて視覚的にわかりやすく解説しました。前回あまり詳しく説明されなかった効果音モードや映像同期モードは図を用いたり具体的な例を挙げて解説したので，すぐに自作のアプリケーションに応用できることと思います。

また，Z-MUSICの内部ワークを完全公開し，ワークひとつずつに解説を入れてあります。今回の特集記事にもありますが，このワークを参照，書き換えることでより高度なアプリケーションの実現が可能です。

ページ数がV1の2倍近くの厚いマニュアルになってしまいましたがV1より確実に理解しやすくなっているはずです。

A ♪sound mind in a sound body

ずいぶんと新機能が追加されましたので，前バージョン以上にマニアックなものになりました。最初から全部の機能を使いこなそうとは思わず，人の演奏データを聴いたり，打ち込んだり，既存データのアレンジなどをしたりしつつ，徐々にZ-MUSICに慣れていってください。

では，Z-MUSICシステムがあなたの発想のペンに，X68000がキャンバスになりますように。

[特集]
Z-MUSICシステム
ver. 2.0
▼
Z-MUSIC支援ツール

これが新しいZPCNV.Rなのだ

Tateno Mitsuru 舘野 暢

新しくなったのはシステム本体だけではありません。強化されたサポートツールのなかから，ZPCNV.Rを取り上げてみました。AD PCMストリングスを使ったサンプル曲を題材にZPCNV.Rの使い方を解説します。

あ，どうもはじめまして。私，舘野暢という新参者なんです。10月号のPassing Breezeでお世話になりました。覚えてます？　今回はZPCNV.Rの解説をやらせてもらうことになりました。新しいZ-MUSICではZPCNV.Rも大幅に強化されています。ここでは拡張された機能を使ったサンプルを題材に使い方を紹介します。

まずデータを入力するところから，とりあえず曲が聞けるようになるまでを解説してみたいと思います。わかりやすくしたつもりなので聞けるようになるまでがんばってくださいね。

過去になんらかのかたちでZPCNVがどんなものなのかわかっているのなら，コマンドの解説に飛んでかまいません。

なにをするもの？

さてZPCNVとはどんなものでしょう。ひと言でいえばPCMの統合加工ツールなんです。これひとつで一応なんでもできます。なかなか奥が深くてエライものなんです。

では，そのZPCNVでなにをするんでしょう。そ，それはZPDファイルを作るんです。それじゃ，ZPDってなぁに？　演奏に必要なドラムキット&ギターなどをあらかじめAD PCMデータから加工して用意しておくものですね。

Z-MUSICをよく知らないという人でもLIVE inのデータで曲のリスト以外に変なリストが載っているのを見たことがあると思います。これは「AD PCMコンフィギュレーションファイル」というものですが，その曲で使うAD PCMデータの塊（ZPDファイル）を作るためのものなんです。

必要なもの

ZPCNV.RとZMUSIC.XとPCM8.Xとちょっとした知識さえ揃っていれば，最近ハヤリの(?)「PCMがオラオラのガガーリン」な曲データが誰にでも作れてしまうわけなんです。あ，でもAD PCMデータがないとSTRやドラムが鳴らないのは絶対確実なので，この際Z-MUSICシステムver. 2.0を手に入れましょう。

この中にはAD PCMデータがウハウハ状態なので，この先ZMUSICerには必須アイテムでしょう。あとデータを入力するためのエディタが必要です（そりゃそうだ）。

さっそく入力

とりあえず，サンプルです。題材は「スターブレード」のネーミング曲です。

このゲームは大阪の花博に出典されたギャラクシアン[3]筐体を小型化した感じのポリゴンシューティングです。初めてプレイしたときはそのグリグリ動く画像に感動しましたよ……。ラストの敵を倒したあとのエンディングへの展開は涙チョチョ切れでした。まだ見てない人はコンティニューを数回やってでも見る価値はあるでしょう。

では，リスト1を「StarB4.CNF」としてテキストエディタで全部入力してみてください。ちゃんとセーブしてくださいよ。ちなみに，ZMUSIC.XとZPCNV.Rは付録ディスクについていたもの，AD PCMデータはZ-MUSICシステムver.1のときのものだけで演奏可能です。

続いてリスト2をエディタで打ち込んでください。

ZPCNVでZPDにコンバートするわけなんです。書式はコマンドシェル上で「ZPCNV －スイッチ　CNFファイル名(.CNF)」です。ここでは，

A>ZPCNV-D StarB4.CNF

と打ち込みます。-Dスイッチをつけると，コンバート中に問題があればメッセージにその行番号を示してくれるので，デバッグが楽になります。

続いてZMSファイルをコンパイルしてみます。

A>ZMUSIC-C StarB4.ZMS

エラーが出たら，表示された行番号の行を調べてください。

さて，これでZPDができたハズです。ちゃんとできてますか？　無事にできていたらさっそく鳴らしてみるとしますか。

演奏データを聞くには

すべてのZ-MUSICのデータにいえることですが……，ZMUSIC.Xが常駐していないと鳴りません。頑張って常駐しようとしても「常駐できないっていわれちゃうんだなぁ」なーんていう人は，だいたいほかの音楽ドライバがすでに常駐してるクサイのでCONFIG.SYSやAUTOEXEC.BATをよく確認してくださいね。

ちなみに常駐には，

A>PCM8
A>ZMUSIC

とするだけです。ver.2.0ではトラックバッファが標準で128Kバイト，AD PCMバッファが256Kバイト指定されてますから，普通に使う分には特別なオプションスイッチ指定は必要なくなりました。ということで，Z-MUSICでデータを作成する人はなるべくこの範囲で収まるようにしましょう。

で，常駐できたなら，あとは聞いてみるだけですね。

A>ZP StarB4.ZMS

と打ち込んでください。どうです，ちゃんと鳴りましたか？

コマンドの解説

さて，話が変わります。そう，本題のCNF内のコマンド説明です。実はそんなにたいしたことはやってません。ドラムのPCM指定と，ストリングスその他の設定です。

まずはドラムのほうの概念です。

鍵盤楽器を思い浮かべてください。ピアノでもオルガンでもなんでもいいです。ピアノでオクターブ4のドを弾くと「ピアノのド」が鳴りますね。これが普通です。

しかし，ドラムキットでは「ピアノのド」でなくてもいいのです。どういうことかというと，StarB4.CNFの16行目からの,

10　　　= HIQ.PCM,v02
.o2e　　= HIQ.PCM,v20,m10,618

というのはオクターブ2のミを弾いたらHIQを「ピッ」と鳴らすという設定です。37行まではすべてドラムの設定なのです。つまり，音階を持たない打楽器を表す場合，鍵盤の1つひとつになんでも好きな音を割り当てることができるのです。素直にPCMを割り当てていないので見づらいでしょうが，やっていることは簡単です。

上の例を説明します。まずv20というのは元になるPCMデータを100%とし，20%の音量で出力するコマンドです。要するに音量を下げてます（PCM8.Xを組み込むことでAD PCMチャンネルでも音量指定が可能になります）。

あ，なぜ音量を下げたのでしょうか？　それはZMS中のPCMトラックのボリュームをv9に統一したからです。では，なぜv9なのでしょうか？　実はv9というのはPCM8にとって楽な音量なんです。というかv9が基本音量なわけでv10やv15とするとPCM8が音量を大きくしてから再生処理をするために若干スピードが落ちます。重そうな曲ではなるべくv9でデータを作ってください。

次にv20の後ろに「m10,618」とあります。おっと，その前に,

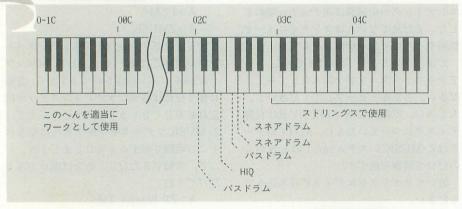
10　　　= HIQ.PCM,v02

がありますね。ここの行頭は「o4c」などではなく「10」ですね。10とはなんでしょう。ここの10はZ-MUSICの仮想鍵盤のいちばん低い音から10度上の鍵盤という意味です。普段使わないような音域なので作業用音階として適当に「10」を使っているわけです。要するに，いったんノート10を使ってPCMを加工しておいてあとから.o2eの音と混ぜてやるわけです。

「618」の部分は.o2eと10を混ぜるときに10を618バイト遅らせて混ぜるときの指定です。10のほうが音量も発音タイミングも遅いのでエコー効果を出しているわけです。

次にストリングスのほうにいきましょ。実はこっちのほうが簡単かもしれません。単に対象になる鍵盤に音程をあわせて設定すればいいからです。さっきのときみたいに「o2eはピッ」と覚えなくていいのです。ドラムキットは好きなところに好きな音を割り当てられるのですから，音階に合致した音を並べておけば普通の楽器のようにも扱えるというわけです。

ZPCNVではAD PCM音に音階をつけることができます。音程の指定は元PCMを基本として「何度分音程を上げ下げするか」で行います。具体的には,

.o3c　= 1,p-12,c0,16000,v70

のように書きます。

説明するとノート1のPCMを70%の音量（v70）でファイルの先頭（0バイト目）から16000バイト以内を使用し（c0，16000），1のPCMから12度下げた音程（1，p−12）に設定しています。12度というのはちょうど1オクターブ下げたことになります。ちなみにドからソに上げたいときは7度上げることになるので，p7と書くことになります。

あ，なんでc0，16000なんてするのかというとp-12としたとき，変換後のAD PCMデータの大きさが元の2倍になってしまうので……，イヤすぎでしょ？

ver.2.0になって

もともとAD PCMデータの設定やミキシングが主目的であったZPCNVですが，ver.2.0では本当にすごいことになっています。PCMが8トラックすべてステレオ出力可能なハードだったら大変なことになっていたかもしれません。

実はver.2.0ではPCM加工することにより元PCMから@K1したようなものまで出力します。つまりAD PCMディチューンができるということです。次期X68000は多チャンネルステレオPCMになっていてほしいところです。イヤ，マジで。

さらにAD PCMでは普通考えられなかったオクターブ4のドからオクターブ3のミにポルタメントするといったこともできるようになっています。うまく使いこなせば，リアルなギター音や新しい音色の発見ができるかもしれませんね。ドラムをポルタメントさせてしまうとか。

＊　　＊　　＊

どうも急ぎ足になってしまいましたが，1コマンドずつ実験しながらいくとわかりやすいのでは？　調子に乗って自分の声をサンプリングしてドラムキットとして鳴らしてみたりすると，近所にかなり怪しい噂が流れ出す可能性があるので気をつけてください。次回があれば，もっとえらそうなデータを作ってみたいですね。

図1　Z-MUSICの仮想鍵盤

リスト1

```
 1: /-----------------------------------------------------------
 2: /  StarB4.CNF -> StarB4.ZPD
 3: /
 4: /  for X680x0 PCM8.X & ZMUSIC.X
 5: /
 6: /  STARBLADE(c)namco 1991   Blue Flight
 7: /
 8: /  Programmed by Tateno Mitsuru 舘野 暢
 9: /               MIYA0348 TTN 1992/07/05,1993/11/15+
10: /
11: /-----------------------------------------------------------
12: /
13: /  ZPCNV -d StarB4.CNF
14: /
15: /     ピ
16: 10 = HIQ.PCM,v02
17: .o2e       = HIQ.PCM,v20,m10,618
18:
19: /
20: /     B.D.
21: 14 = CASTA.PCM,v20
22: 4  = WDK1.PCM,v25,m14
23: 6  = KICK11.PCM,v30,m4
24: 7  = 6,v15
25: .o2c         = 6,m7,618
26: 8  = KICK11.PCM,v40
27: 9  = KICK11.PCM,v10
28: .o2f         = 8,m9,618
29:
30: /
31: /     S.D.
32: 5  = HIGHTB.PCM,v23
33: 11 = FNKS_.PCM,p-2,m5,v58
34: 12 = WHIP1.PCM,m11,v4
35: 13 = 12,v15
36: .o2g         = 12,m13,618
37: .o2g-        = .o2g,v70
38:
```

▶コナミの次回作は，開発コードNo.の順番からして，〇ラIIIでは……。
黒部　浩孝(20)三重県

```
39: /
40: /    STR
41: 1  = STR.PCM,v22,c315
42: .o3c         = 1,p-12,c0,16000,v70
43: .o3f         = 1,p-7,c0,16000
44: .o3f#        = 1,p-6,c0,16000
45: .o3g#        = 1,p-4,c0,16000
46: .o3a         = 1,p-3,c0,16000
47: .o3a#        = 1,p-2,c0,16000
48: .o4c         = 1,p00,c0,16000
49: .o4c#        = 1,p01,c0,16000
50: .o4d         = 1,p02
51: .o4d#        = 1,p03
52: .o4f         = 1,p05
53: .o4f#        = 1,p06
54: .o4g         = 1,p07
55: .o4g#        = 1,p08
```

```
56: .o4a#        = 1,p10
57:
58: /----------------
59:
60: .erase 14
61: .erase 13
62: .erase 12
63: .erase 11
64: .erase 10
65: .erase 9
66: .erase 8
67: .erase 7
68: .erase 6
69: .erase 5
70: .erase 4
71: .erase 1
```

リスト2

```
  1: .comment -STARBLADE- Blue Flight (C)namco  By TTN 92/07/05,
93/11/20+ (+PCM8)
  2: /------------------------------------------------------------
  3: /  StarB4.ZMS
  4: /
  5: /  for X680x0 PCM8.X & ZMUSIC.X
  6: /
  7: /  STARBLADE(c)namco 1991   Blue Flight
  8: /
  9: /  Programmed by Tateno Mitsuru 舘野 暢
 10: /                MIYA0348 TTN 1992/07/05,1993/11/15+
 11: /
 12: /------------------------------------------------------------
 13:
 14: /------------------------------
 15: /    INIT
 16:
 17: (b0)
 18: (i)
 19: (o125)
 20:
 21: /------------------------------
 22: /    ADPCM BLOCK DATA setup
 23:
 24: .adpcm_block_data = StarB4.ZPD
 25:
 26: /------------------------------
 27: /    Track setup
 28:
 29: (m1,1000)(aFM1,1)
 30: (m2,1000)(aFM2,2)
 31: (m3,1000)(aFM3,3)
 32: (m4,1000)(aFM4,4)
 33: (m5,1000)(aFM5,5)
 34: (m6,1000)(aFM6,6)
 35: (m7,1000)(aFM7,7)
 36: (m8,1000)(aFM8,8)
 37: (m09,1000)(aadpcm,09)
 38: (m10,1000)(aadpcm,10)
 39: (m11,1000)(aadpcm,11)
 40: (m12,1000)(aadpcm,12)
 41: (m13,1000)(aadpcm,13)
 42:
 43: /------------------------------
 44: /    OPM DATA setup
 45:
 46: / Melo.2
 47: (@69,    18,  0,  0,  7,  0, 48,  0,  4,  0,  0,  0
 48:          17,  0,  0,  7,  0, 44,  0,  1,  0,  0,  0
 49:          16,  0,  0,  7,  0,  6,  0,  1,  0,  0,  0
 50:          18,  0,  0,  7,  0,  9,  0,  1,  0,  0,  0
 51:           6,  0, 15)
 52: / Melo.1
 53: (@70,    21,  9,  0,  7,  2, 45,  0,  2,  0,  0,  0
 54:          21,  9,  0,  7,  2, 45,  0,  2,  0,  0,  0
 55:          21,  7,  2,  7,  3, 49,  0,  0,  0,  0,  0
 56:          20,  9,  0,  7,  1,  0,  0,  1,  0,  0,  0
 57:           3,  7, 15)
 58: / Melo.1'
 59: (@71,    21, 12,  0,  7,  2, 45,  0,  2,  0,  0,  0
 60:          21, 12,  0,  7,  2, 43,  0,  2,  0,  0,  0
 61:          21, 10,  2,  7,  3, 40,  0,  0,  0,  0,  0
 62:          21, 12,  0,  7,  2,  0,  0,  1,  0,  0,  0
 63:           3,  7, 15)
 64: / E.Bass
 65: (@72,    28, 13,  6,  5,  1, 32,  0,  0,  0,  0,  0
 66:          28, 13,  6,  5,  1, 30,  2,  4,  0,  0,  0
 67:          28, 13,  6,  5,  1, 40,  0,  0,  0,  0,  0
 68:          28, 14,  6,  6,  1,  6,  0,  1,  0,  0,  0
 69:           2,  7, 15)
 70: / H.H c
 71: (@73,    31, 13, 12,  5, 13,  4,  0, 15,  3,  1,  0
 72:          26, 17,  9,  5, 10,  4,  0, 10,  3,  2,  0
 73:          25, 17,  9,  5, 10,  4,  0,  3,  0,  3,  0
 74:          31, 18, 12,  7, 10,  6,  0,  1,  6,  0,  0
 75:           2,  7, 15)
 76: / H.H o
 77: (@74,    31, 13, 12,  5, 10,  0,  0, 15,  3,  1,  0
 78:          26, 16, 10,  5,  3,  0,  0, 10,  3,  2,  0
 79:          25, 16, 10,  5,  3,  0,  0,  3,  0,  3,  0
 80:          31, 16, 13,  7,  3,  2,  0,  1,  6,  0,  0
 81:           2,  7, 15)
 82: / Thu.
 83: (@75,    20,  2,  0,  5,  2, 32,  0,  1,  0,  0,  0
 84:          20,  2,  0,  6,  2,  7,  0,  1,  0,  0,  0
 85:          20,  2,  0,  5,  2, 17,  0,  0,  0,  0,  0
 86:          20,  2,  0,  6,  2,  0,  0,  1,  0,  0,  0
 87:           4,  7, 15)
 88: / Thu.2
 89: (@76,    20,  4,  7,  5,  3, 32,  0,  1,  0,  0,  0
 90:          20,  4,  7,  6,  3,  7,  0,  1,  0,  0,  0
 91:          20,  4,  7,  5,  3, 17,  0,  0,  0,  0,  0
 92:          20,  4,  7,  6,  3,  2,  0,  1,  0,  0,  0
 93:           4,  7, 15)
 94: / +PCM
 95: (@77,    17,  0,  0, 15,  1, 34,  0,  1,  0,  0,  0
 96:          17,  0,  0, 15,  1, 36,  0,  2,  0,  0,  0
 97:          17,  0,  0, 15,  0, 34,  0,  1,  0,  0,  0
 98:          17,  0,  0, 15,  0,  3,  0,  2,  0,  0,  0
 99:           2,  7, 15)
100: / ウラ
101: (@78,    15,  0,  0,  7,  0, 47,  0,  1,  0,  0,  0
102:          15,  0,  0,  7,  0, 47,  0,  1,  0,  0,  0
103:          15,  0,  0,  7,  0, 32,  0,  1,  5,  0,  0
104:          15,  0,  0,  7,  0,  3,  0,  2,  1,  0,  0
105:           4,  7, 15)
106:
107: /------------------------------
108: /    MML DATA setup
109: /----------------
110: / OPM
111:
112: / Melo.
113: (t1)  r1r6@v121o5164p3 @H36,0 S2 @S6,4 @M06 @A03 @K04 q8
114: (t1)  @71(d,f)&f1&f4....@v120@69(d-,d)&d8...(e,f)&f8...
115: (t1)  f1(>b<c)&c4....@v121{cde-}2@v123(g-,g)&g1&g4....
116: (t1)  (d,e-)&e-8...(g-,g)&g8...d2(d,e-)&e-4....(e,f)&f4....
&
117: (t1)  @v121{fga-}2@v123(a,b-)&b-2.....(c,d-)&d-2^8...
118: (t1)  (a,b-)&b-8...a-1(>b<c)&c2.....(f,g-)&g-4....f2e-2d-2
119: (t1)  (c,d-)&d-2.....(g,a-)&a-2.....
120: (t1)  [do]>(a,b-)&b-4....<c2(d,e-)&e-2^8...@v121{dcd}4>
121: (t1)  @v123(a,b-)&b-2.....&b-1(a,b-)&b-4....<c2(d,e-)&e-4..
..
122: (t1)  @v121{dcd}2@v123(>b-<c)&c2.....&c1
123: /
124: / 以下 114行～123行と ほぼ同じ(主に"@v～"を消す)
125: (t2)  r1  @v127o5164p3 @H49,0 S2 @S6,4 @M12 @A03 @K02 q8
126: (t2)  @71(d,f)&f1&f4....      @70(d-,d)&d8...(e,f)&f8...
127: (t2)  f1(>b<c)&c4....       {cde-}2      (g-,g)&g1&g4....
128: (t2)  (d,e-)&e-8...(g-,g)&g8...d2(d,e-)&e-4....(e,f)&f4....
&
129: (t2)        {fga-}2      (a,b-)&b-2.....(c,d-)&d-2^8...
130: (t2)  (a,b-)&b-8...a-1(>b<c)&c2.....(f,g-)&g-4....f2e-2d-2
131: (t2)  (c,d-)&d-2.....(g,a-)&a-2.....
132: (t2)  [do]>(a,b-)&b-4....<c2(d,e-)&e-2^8...      {dcd}4>
133: (t2)        (a,b-)&b-2.....&b-1(a,b-)&b-4....<c2(d,e-)&e-4..
..
134: (t2)        {dcd}2      (>b-<c)&c2.....&c1
135:
136:
137: / E.Bass
138: (t3)  r1@72@v122o2112p3 @K07 q8 @q8
139: (t3)  |:12b-<b->:||:12a<a>:||:12a-<a->:||:12g<g>:|
140: (t3)  |:12g-<g->:||:12f<f>:||:12e-<e->:||:6g-<g->:|
141: (t3)  |:3a-<a->:|b-<c@v126e-a-@v127<ce->>@v124
142: (t3)  |:3 |:12b-<b->:|||:12a-<a->:||:12g<g>:||:12a-<a->:|:|
 ;$cc,7
143: (t3)                    |:12a-<a->:||:12g<g>:||:12a-<a->:|
144: (t3)  [do]|:12b-<b->:| |:12a-<a->:||:12g<g>:||:12a-<a->:|[l
oop]
145:
146:
147: / Thu.
148: (t4)  r1@v110o211p1 @K-6 q8 r64.
149: (t4)  @76b-&b-2f4b-4@75a&a @76a-&a-g&g
150: (t4)  @75g-&g-f&fe-&e-g-a-
151: (t4)  [do]@76b-&b- a-&a- g&g a-&a-
152: (t5)  r1@v110o211p2 @K06 q8
153: (t5)  @76b-&b-2f4b-4@75a&a @76a-&a-g&g
154: (t5)  @75g-&g-f&fe-&e-g-a-
155: (t5)  [do]@76b-&b- a-&a- g&g a-&a-
156:
157:
158: / H.H
159: (t6)  @73@v102o3112p3 @K14 q8
160: (t6)  |:4@74c@73cc:|
161: (t6)  [do]@74c@73cccccccc@74c@73cc
162:
163:
164: / +PCM
165: (t7)  @73@v083o3112p1 @K00 q8
166: (t7)  r24|:3@74c@73cc:|@74c@73cc24
167: (t7)  @77@v110o412p3 @H00,0 S1 @S5,4 @M08 @A04 @K04 r32.
168: (t7)  rde-ff1c1&v08c4 @v110e-4f4g4ge-4g411df
169: (t7)  @v111p2b-&b-a-&a- g-2f2e-d-e-
170: (t7)  [do]b-v07b-2@v111b-2 b-&b- g&g a-&a-
171:
172:
173: / ウラ,+PCM
174: (t8)  @78@v110o411p3 @H00,0 S1 @S5,4 @M10 @A04 @K03
175: (t8)  r1(e-64,f)&f2.....&fa&aa-&a-g&g
176: (t8)  @77           p1                   @M08 @A04 @K05 r32.
177: (t8)  f&fe-&e- d-&d- >b-<d-2c2
178: (t8)  [do]o4df a-&a- e-&e- e-&e-
179:
180:
181: /----------------
182: / PCM
183:
184: / STR
185: (t9)  @1o4v912 r1
186: (t9)  rde-fffc1v7c16v4c8. v9e-4f4g4ge-4g4ddff
187: (t9)  b-b-b-b- a-a-a-a- g-fe-e- d-d-e-e-
188: (t9)  [do]b-b-_36b-~b- b-b-b-b- gggg a-a-a-a-
189: /
190: (t10) @1o3v912 r1
191: (t10) rb-b-1a1a1v7a16v4a8. v9<c4c4e-4e-c4e-4>b-1<dd
192: (t10) ff.f.e-e-.e-.d-d-.d-.>b-1<d-cv9
193: (t10) [do]o4ddff a-a-a-a-|:8e-:|
194: /
195: (t11) @1o3v912 r1
196: (t11) rff1c1f1v7f16v4f8. v9a-4a-4<c4c>a-4<c4>f1b-1
197: (t11) v8<d-.d-.d-c1c1> b-1b-1 v9g-1a-1
198: (t11) [do]b-1b-1<e-.e-.e->b-1b-1<c1c1
199:
200:
201: / B.D.
202: (t12) @1o2112v9rc_3c6~c_c6~c_c6~c_c~
203: (t12) [do]c_c6~ee6c6ce4c_c6~e4c6ceff
204:
205:
206: / S.D.
207: (t13) @1o2112v9e4e4e4eg-g4r
208: (t13) [do]|:7g4r4:|g-g4r6
209:
210: /------------------------------
211:
212: (t1)  [loop]
213: (t2)  [loop]
214: (t3)  [loop]
215: (t4)  [loop]
216: (t5)  [loop]
217: (t6)  [loop]
218: (t7)  [loop]
219: (t8)  [loop]
220: (t9)  [loop]
221: (t10) [loop]
222: (t11) [loop]
223: (t12) [loop]
224: (t13) [loop]
225:
226: /------------------------------
227: /    PLAY
228:
229: (p)
```

▶ナムコの「リッジレーサー」には，本当にビリビリッときました。ポリゴンに興味のある私にはいい刺激になりました。早く実物を見てみたいです。ナムコ万歳！

田口　英司(17)茨城県

［特集］
Z-MUSICシステム
ver. 2.0

サポートツールZMI.X

曲データの演奏と管理

Sudo Yoshimasa 須藤 芳政

Z-MUSICシステムの初心者に送る「Z-MUSICでのトラブルシューティング講座」です。またZ-MUSICでのデータ管理に便利なツールもあわせて紹介します。

Z-MUSICをつい最近使い始めたばかりの方で「演奏データを手に入れたけど演奏できない，手元には付録についてきた'ZMUSIC.X'と'PCM8'だけでほかにはなんも持っていないし」なんて状態に陥って混乱している人はいませんか？

そこで今回のテーマは「曲データを演奏させられる人間になる」です。曲データを演奏させる場合においてデータ側に異常があれば当然演奏できませんが，そうではないのに音が鳴らないという問題にぶつかって苦しみ，ひとりだけ周囲から遅れをとってしまうのではないかと不安に駆られている人を対象として話を進めていきます。悩む前に実践だ！

ACT1 Z-MUSICを組み込む

まず用意するのはZ-MUSICの本体である'ZMUSIC.X'のファイルです。この時点で「え？」と思ってしまった人はいませんね？用意ができたらコマンドラインから，

　ZMUSIC -T100 -P200

と打ってみましょう。

スイッチ-Tの後ろにある100の数字は演奏データを格納するトラックバッファのサイズで単位はKバイトです。これが小さすぎると曲データによっては演奏ができないものが出てきます。スイッチ-Pの後ろにある200の数字はドラムなどのリズム楽器に使われるAD PCMサンプリングデータを格納するPCMバッファのサイズで単位はトラックバッファと同様にKバイトです。

そしてリターンキーを押してZ-MUSICのタイトルが出てくれば組み込み完了です。この程度のバッファサイズ指定していて，こんなことはまず起こらないと思うのですが，もし“ERROR”と表示されたらメモリが不足しているので，指定するトラックバッファとPCMバッファのサイズをそれぞれ減らし（例：-T50 -P100)，再びトライしてみましょう。

ちなみに，なにもつけずに，

　ZMUSIC

と常駐させると，トラックバッファを128Kバイト，PCMバッファを256Kバイト確保します。バッファの使用量は小さく指定することもできるので，メモリが足りなくなったときは上記のように対応してください。

ACT2 とりあえず演奏させてみる

それでは演奏データのファイルを用意してください。拡張子が'.ZMS'のファイル(以下，ZMSファイル)または'.ZMD'のファイル（以下，ZMDファイル）であることはもう皆さんご承知ですね。「ええ？ なにそれ」と思ってしまった人のために簡単に説明すると，ZMSファイルはMML(ミュージックマクロランゲージ）で表記されたテキスト形式のファイルです。

ですからZMSファイルをスクリーンエディタで呼び出せば毎月Oh!Xに掲載されている曲データリストから行番号を取り除いたものとそっくりなものが表示されます。

もうひとつのZMDファイルはZMSファイルをコンパイルして生成されたオブジェクトファイルです。組み込まれているZ-MUSICと演奏するZMDファイルを生成したZ-MUSICのバージョンが異なると暴走する危険があるので注意しましょう（これは演奏データに異常があることと同様なので今回は触れません)。コンパイルの方法はあとで述べますが，ZMDは「スクリーンエディタで見られないファイル」ぐらいに考えておいて結構です。

さてさて，ZMSまたはZMDファイルを'OPM'というファイルにコピーすると演奏スタートです。たとえば'VENOM.ZMS'を演奏させたいのならばコマンドラインから，

　COPY VENOM.ZMS OPM

と打ってリターンを押すだけです。「そんなことしたら'OPM'ってファイルができちゃうじゃないか」などという心配はありません（Z-MUSICを組み込んでいればの話)。

ACT3 なぜ演奏できないの？

ちゃんと演奏されましたか？ まったく音が鳴らなかった，演奏が変だ，暴走してしまったという人のためにこれからいろいろ原因を調べていきましょう。

・音がまったく鳴らない

1)　トラックバッファが不足している

もう一度ACT1に戻ってトラックバッファをもっと大きく設定しなおしましょう。組み込み直す前に，

　ZMUSIC -R

で一度ZMUSIC.Xの常駐を解除するのも忘れずに。

2)　MIDI音源対応だった

MIDI音源をつないでいないのにMIDI音源のみに対応したデータを演奏すれば当然なにも聞こえませんよね。解決方法は「音源を買うまで我慢する」でしょうか。

3)　その他（論外！）

本体のボリュームつまみ，あるいは外部へ接続したアンプのボリュームが0になっている場合です。まあ，こんな失敗はめったにやらかしませんが。

・音は出ているがなにか変だ

1)　MIDI音源の機種が違う

たとえばCM-64用のデータをSC-55で演奏させてしまったときなどです。

2)　AD PCM演奏に必要なファイルが見つからない

現在，内蔵音源でドラムといえばAD PCMで鳴らすのが常識といわれるくらいになっています。サンプリングデータをまとめたZPDファイル(拡張子が'.ZPD')，またはサンプリングデータをZ-MUSICへ1つひとつ渡すためのAD PCMコンフィギュレーションファイル（拡張子は一般に

'.CNF')が演奏データ側から指定されている場合に，そのファイルを見つけることができなかったらほかのFM音源やMIDI音源は鳴っていてもAD PCMは鳴っていないという状態になります。

3) PCMバッファが不足している

Z-MUSIC組み込み時に指定したPCMバッファのサイズが小さくてAD PCMデータの登録ができなかった場合にも2)と同じ症状が出ます。ACT1に戻ってPCMバッファを設定しなおしましょう。

4) PCM8が組み込まれていない

サンプリング音が途切れてどう考えても変な演奏に聞こえるとき，これはPCM8(PCMを疑似的に8チャンネル同時発音させるアプリケーション）を組み込む必要があるデータであると思われます。一度Z-MUSICの常駐を解除してから'PCM8.X'のファイルを用意して実行するとPCM8が常駐するので，そのあとに再びZ-MUSICを組み込んでみてください。

ファイル情報を表示してみよう

演奏させるファイル内で外部のなんというファイルを指定しているのか，トラックはいくつなのか，そして演奏デバイス（音源ですね）はなにを必要とするのか？ そんな情報を表示するプログラムを作ってみようということになりました。

今回はとりあえずZMDファイルの情報をある程度表示できるまでの簡単なプログラムなので，残念ながらZMSファイルの情報は得られません。

ZMSファイルをコマンドラインから，

ZMUSIC -C ファイル名

でコンパイルしてZMDファイルを作ってからこのプログラムを実行しましょう（このコンパイル時にエラーが発生したらZMSファイルに問題があります)。こんなことをするくらいならZMSファイルを直接スクリーンエディタで覗いたほうが手っ取り早いのですが，ZMSのコマンドがわからない人は試しにこっちを使ってみてください。

使い方はコマンドラインから，

ZMI

と打ってリターンキーを押すだけ。ディレクトリ内にあるすべてのZMDファイルの情報をZMDファイル名，総トラック数，ZPDファイル名，CNFファイル名，MDDファイル名，演奏デバイスの順に表示します。

MDDファイルはここで初めて登場しますが，MIDI楽器へ送信するデータがひとまとめにしてあるファイルです。もしも，

ZMI -C

としたならば各ファイルのコメントを表示します。たいていの演奏データはこのコメント部分に曲名，作者が書き込まれているのです。

-Cのスイッチありとなし両方の情報を得たいときには，

ZMI ファイル名

で表示可能です。しかし複数のファイルに対しては一度に実行できません。

がんばって使いこなそう!

パソコン通信のように次々といろんな情報が得られれば，ぶつかった問題も次々と解決していくのですが，そういう環境に置かれていないとやはりつらいものです。付録ディスクで'ZMUSIC.X'と'PCM8.X'をやっと手に入れた人たちには今後がんばっていただだきたく思います。

参考文献

吉沢正敏 市原昌文，X68000環境ハンドブック，工学社

リスト

```
  1: *****************************************
  2: *       ZMDIR. X
  3: *****************************************
  4:
  5: FNC     macro   number
  6:         dc.b    $ff,number
  7:         endm
  8:
  9: IOCS    macro   number
 10:         move.l  #number,d0
 11:         trap    #15
 12:         endm
 13:
 14: ****************
 15:
 16:         lea     work(pc),a6
 17:         lea     usrsp(pc),sp     *ユーザースタックを設定
 18:
 19:         FNC     $19              *カレントドライブを得る
 20:         add.b   #$41,d0
 21:         move.b  d0,cur_dir-work(a6)
 22:
 23:         pea     path_buf(pc)     *カレントディレクトリゲット!
 24:         move.w  #0,-(sp)
 25:         FNC     $47
 26:         addq.l  #6,sp
 27:
 28:         pea     cur_dir(pc)
 29:         FNC     $09
 30:         addq.l  #4,sp
 31:         bsr     cr_prt
 32:
 33:         move.b  (a2)+,d2         *コマンドラインのバイト数をキャンセル
 34:
 35: switch_check:                    *設定スイッチをチェック
 36:         move.b  (a2)+,d0
 37:         beq     set_flname
 38:
 39:         cmpi.b  #" ",d0
 40:         beq     switch_check
 41:
 42:         cmpi.b  #"-",d0          *スイッチを発見
 43:         bne     set_flname
 44:
 45:         move.b  (a2)+,d0
 46:         and.b   #$5f,d0          *スイッチを大文字に
 47: sw_1:
 48:         cmpi.b  #"C",d0          *コメント表示のスイッチ発見
 49:         bne     chk_lop
 50:         move.b  #-1,flg_c-work(a6)
 51:         bra     chk_lop
 52:
 53: chk_lop:
 54:         move.b  (a2)+,d0
 55:         beq     set_flname
 56:         cmpi.b  #" ",d0
 57:         beq     switch_check
 58:         bra     chk_lop
 59:
 60: set_flname:
 61:         subq.l  #1,a2
 62:
 63:         move.b  (a2),d0
 64:         beq     wild_mode
 65:
 66:         lea     fl_name(pc),a0   *ファイルネームを登録
 67:         moveq   #22,d0
 68:         clr.b   d2
 69: set_flop:
 70:         move.b  (a2)+,d1
 71:         cmpi.b  #".",d1
 72:         bne     set_flopj1
 73:         move.b  d1,d2
 74: set_flopj1:
 75:         move.b  d1,(a0)+
 76:         beq     set_flopj2
 77:         dbra    d0,set_flop
 78:
 79: set_flopj2:
 80:         tst.b   d2
 81:         bne     set_flopj3
 82:         move.b  #".",-1(a0)
 83:         move.b  #"Z",(a0)
 84:         move.b  #"M",1(a0)
 85:         move.b  #"D",2(a0)
 86:         move.b  #0,3(a0)
 87:
 88: set_flopj3:
 89:         bsr     fl_open
 90:         tst.b   d0
 91:         bne     prt_err_op
 92:
 93: zm_info:
 94:         bsr     get_para
 95:
 96:         tst.b   flg_z-work(a6)   *ZMDファイルなのかチェック
 97:         bne     not_zmd
 98:
 99:         bsr     line_prt2        *ファイル名を指定したときの処理
100:
101:         pea     para1_txt(pc)
102:         FNC     $09
103:         addq.l  #4,sp
104:
105:         pea     para2_txt(pc)
106:         FNC     $09
107:         addq.l  #4,sp
108:
109:         bsr     line_prt
110:
111:         bsr     fln_trans
112:         move.l  #".ZMD",fln_txt_k-work(a6)
113:
114:         pea     fln_txt(pc)
115:         FNC     $09
116:         addq.l  #4,sp
117:
118:         cmp.b   #-1,comment_buf-work(a6)         *コメント表示
119:         beq     zm_infl
120:
121:         bsr     prt_comm
122:
123: zm_infl:
124:         bsr     cr_prt
125:         bsr     line_prt2
126:
127:         pea     para4_txt(pc)
128:         FNC     $09
129:         addq.l  #4,sp
130:
131:         bsr     line_prt
132:
133:         bsr     prt_fln_s
134:
135:         bsr     line_prt2
136:         IOCS    $00
137:         FNC     $00              *終了
138:
139: wild_mode:                       *ファイル名を指定しなかった場合
140:         bsr     line_prt2
141:
142:         pea     para1_txt(pc)    *表示するパラメータ名を選択して表示
143:         FNC     $09
144:         addq.l  #4,sp
145:
146:         tst.b   flg_c-work(a6)
147:         beq     wil_l1
148:
149:         pea     para2_txt(pc)
150:         FNC     $09
151:         addq.l  #4,sp
152:         bra     wil_l2
153:
154: wil_l1:
155:         pea     para4_txt(pc)
156:         FNC     $09
157:         addq.l  #4,sp
158: wil_l2:
159:         bsr     line_prt
```

```
160:         bra     wild_m_lop
161:
162: wild_m_lop:
163:         bsr     search_file
164:
165:         tst.b   d0
166:         beq     prt_flname
167:
168:         cmpi.b  #-1,d0
169:         beq     prt_nozmd
170:
171:         bsr     line_prt2
172:
173:         FNC     $00                  *終了
174:
175: prt_flname:                          *ファイルネームの表示
176:         bsr     get_para
177:
178:         bsr     fln_trans
179:         move.l  #".ZMD",fln_txt_k-work(a6)
180:
181:         pea     fln_txt(pc)
182:         FNC     $09
183:         addq.l  #4,sp
184:
185:         tst.b   flg_z-work(a6)
186:         beq     wil_l11
187:
188:         pea     message4(pc)         *ファイル異常の表示
189:         FNC     $09
190:         addq.l  #4,sp
191:         bsr     fl_close
192:         bra     wild_m_lop
193:
194: wil_l11:                             *-C使用ならコメントを表示
195:         tst.b   flg_c-work(a6)
196:         beq     wil_l13
197:
198:         cmp.b   #-1,comment_buf-work(a6)
199:         beq     wil_l12
200:         bsr     prt_comm
201:
202: wil_l12:
203:         bsr     cr_prt
204:         bsr     fl_close
205:         bra     wild_m_lop
206:
207: wil_l13:                             *スイッチなし
208:         bsr     prt_fln_s
209:         bsr     fl_close
210:         bra     wild_m_lop
211:
212: prt_nozmd:                           *ZMDが見つからないの表示
213:         pea     message3(pc)
214:         FNC     $09
215:         addq.l  #4,sp
216:         bsr     line_prt2
217:         FNC     $00
218:
219: not_zmd:                             *ZMDではないの表示
220:         pea     message5(pc)
221:         bra     pro_end
222:
223: prt_err_op:                          *指定ファイルがオープンできない
224:         pea     message2(pc)
225:         bra     pro_end
226:
227: pro_end:
228:         FNC     $09
229:         addq.l  #4,sp
230:         FNC     $00
231:
232: ************************************************************
233: *       サブルーチンがいっぱい
234: ************************************************************
235:
236: search_file:
237:         move.b  fl_buf(pc),d0
238:         bne     n_files
239:
240:         move.w  #$20,-(sp)           *ZMDファイルの検索
241:         pea     zmd_wild(pc)
242:         pea     fl_buf(pc)
243:         FNC     $4e
244:         lea     10(sp),sp
245:
246:         cmpi.l  #-2,d0
247:         beq     no_zmd
248:
249: fl_open:
250:         move.w  #0,-(sp)             *ファイルを読み込みオープン
251:         pea     fl_name(pc)
252:         FNC     $3d
253:         addq.l  #6,sp
254:         movea.w d0,a3                *ファイルハンドルをa3レジスタへ
255:         tst.l   d0
256:         bmi     cant_open
257:         clr.l   d0
258:         rts
259:
260: cant_open:                           *ファイルをオープンできなかった
261:         moveq   #-1,d0
262:         rts
263:
264: n_files:                             *2つ以上ファイルが存在する場合
265:         pea     fl_buf(pc)
266:         FNC     $4f
267:         addq.l  #4,sp
268:         cmpi.l  #-2,d0
269:         bne     fl_open
270:         rts
271:
272: no_zmd:                              *ZMDファイルが見つからなかった
273:         moveq   #-1,d0
274:         rts
275:
276: ****************
277: fl_close:                            *ファイルをクローズします
278:         move.w  a3,-(sp)
279:         FNC     $3e
280:         addq.l  #2,sp
281:         rts
282:
283: ****************
284: cr_prt:                              *改行させるだけ
285:         pea     txt_cr(pc)
286:         FNC     $09
287:         addq.l  #4,sp
288:         rts
289:
290: ****************
291: line_prt:                            *ラインを表示するだけ
292:         moveq   #3,d1
293:
294: line_p: pea     line_txt1(pc)
295:         FNC     $09
296:         addq.l  #4,sp
297:         dbra    d1,line_p
298:         bsr     cr_prt
299:         rts
300: ****************
301: line_prt2:                           *ラインを表示するだけその2
302:         moveq   #3,d1
303: line_p2:
304:         pea     line_txt2(pc)
305:         FNC     $09
306:         addq.l  #4,sp
307:         dbra    d1,line_p2
308:         bsr     cr_prt
309:         rts
310:
311: ****************
312: fln_trans:                           *ファイルネームを8文字にしてしまう
313:         moveq   #0,d1
314:         lea     fl_name(pc),a1
315:
316: fln_t1:
317:         move.b  (a1,d1.w),d0
318:         beq     fln_t2
319:         cmpi.w  #9,d1
320:         bcc     fln_t3
321:         cmpi.b  #".",d0
322:         beq     fln_t2
323:         addq.w  #1,d1
324:         bra     fln_t1
325:
326: fln_t2:
327:         move.b  #" ",(a1,d1.w)
328:         cmpi.w  #9,d1
329:         bcc     fln_t3
330:         addq.w  #1,d1
331:         bra     fln_t2
332:
333: fln_t3:
334:         moveq   #9,d2
335:         bsr     txt_cut
336:         lea     fln_txt(pc),a0
337:         lea     fl_name(pc),a1
338:         move.l  (a1)+,(a0)+
339:         move.l  (a1)+,(a0)+
340:         move.b  (a1),d0
341:         beq     fln_t4
342:         move.b  d0,(a0)
343: fln_t4:
344:         rts
345:
346: ****************
347: txt_cut:                             *a1からの文字列をd2文字に切る
348:         moveq   #0,d1
349: txt_c1: move.b  (a1,d1.w),d0
350:         beq     txt_c4
351:         cmpi.b  #$80,d0
352:         bcs     txt_c3
353:         cmpi.b  #$a0,d0
354:         bcs     txt_c2
355:         cmpi.b  #$e0,d0
356:         bcs     txt_c3
357: txt_c2: subq.w  #1,d2
358:         cmp.w   d2,d1
359:         beq     txt_c4
360:         addq.w  #1,d2
361:         addq.w  #1,d1
362: txt_c3: addq.w  #1,d1
363:         cmp.w   d2,d1
364:         bne     txt_c1
365: txt_c4: move.b  #0,(a1,d1.w)
366:         rts
367:
368: ****************
369: prt_comm:                            *コメント表示
370:         moveq   #75,d2
371:         lea     comment_buf(pc),a1
372:         bsr     txt_cut
373:
374:         pea     comment_buf(pc)
375:         FNC     $09
376:         addq.l  #4,sp
377:         rts
378:
379: ****************
380: prt_fln_s:
381:         move.w  num_of_trk(pc),d1       *トラック総数表示
382:         bsr     valp_su
383:
384:         pea     valp_txt(pc)
385:         FNC     $09
386:         addq.l  #4,sp
387:
388:                                         *各種ファイル名表示
389:         lea     zpdfln_buf(pc),a0       *ZPD
390:         move.l  #".ZPD",d3
391:         bsr     prtf_sub
392:
393:         lea     cnffln_buf(pc),a0       *CNF
394:         move.l  #".CNF",d3
395:         bsr     prtf_sub
396:
397:         lea     mddfln_buf(pc),a0       *MDD
398:         move.l  #".MDD",d3
399:         bsr     prtf_sub
400:
401: devp1:                               *使用演奏デバイスの表示
402:         move.b  fm_on(pc),d0         *FM
403:         beq     devp2
404:
405:         pea     fm_txt(pc)
406:         FNC     $09
407:         addq.l  #4,sp
408:
409: devp2:  move.b  pcm_on(pc),d0        *ADPCM
410:         beq     devp3
411:
412:         pea     pcm_txt(pc)
413:         FNC     $09
414:         addq.l  #4,sp
415:
416: devp3:  move.b  midi_on(pc),d0       *MIDI
417:         beq     devp4
418:
419:         pea     midi_txt(pc)
420:         FNC     $09
421:         addq.l  #4,sp
422:
423: devp4:  bra     cr_prt
424:
425: prtf_sub:
426:         lea     fl_name(pc),a1
427:         cmp.b   #-1,(a0)
428:         beq     pr_flspc
429:
430:         moveq   #9,d0
431: pr_sl:  move.b  (a0)+,(a1)+
432:         dbra    d0,pr_sl
433:
434:         bsr     fln_trans
435:         move.l  d3,fln_txt_k-work(a6)
436:         pea     fln_txt(pc)
437:         FNC     $09
438:         addq.l  #4,sp
439:         rts
440:
441: pr_flspc:
442:         pea     fln_spc(pc)
443:         FNC     $09
444:         addq.l  #4,sp
445:         rts
446:
447: ****************
448: get_para:                            *各パラメータをメディアから読み込む
449:         move.b  #0,base_ch-work(a6)
450:         move.b  #-1,comment_buf-work(a6)
451:         move.b  #-1,zpdfln_buf-work(a6)
452:         move.b  #-1,cnffln_buf-work(a6)
453:         move.b  #-1,mddfln_buf-work(a6)
454:         move.b  #0,flg_z-work(a6)
455:         move.b  #0,fm_on-work(a6)
456:         move.b  #0,pcm_on-work(a6)
457:         move.b  #0,midi_on-work(a6)
458:
459:         bsr     one
460:
461:         lea     ZmuSiC(pc),a1        *本当にZMDファイルなのかチェック
462:         moveq   #5,d1
463: getp1:  bsr     one
464:         cmpi.l  #$ff,d0
465:         beq     other_form
466:         move.b  d0,(a1)+
467:         dbra    d1,getp1
468:
469:         cmp.l   #"ZmuS",-6(a1)
470:         bne     other_form
471:         cmp.w   #"iC",-2(a1)
472:         beq     getp2
473:
474: other_form:
475:         move.b  #-1,flg_z-work(a6)
476:         rts
477:
478: getp2:
479:         bsr     one
480: getp3:                               *必要な共通コマンドから情報を得て
481:                                      *それ以外はスキップして読んでいきます
482:         bsr     one
483:
484:         cmpi.b  #$04,d0              *FM音色1
485:         bne     getp4
486:         moveq   #56,d0
487:         bsr     jump
```

▶いまさらながら立体視に凝ってます。年賀状はこれでいこうと思っています。案外できない人もいるみたいですね。
尹　忠秀(24)長崎県

```
488:            bra     getp3
489:
490: getp4:     cmpi.b  #$05,d0             *テンポ
491:            bne     getp5
492:            moveq   #2,d0
493:            bsr     jump
494:            bra     getp3
495:
496: getp5:     cmpi.b  #$18,d0             *MIDIデータ転送
497:            bne     getp6
498:            bsr     one
499:            move.l  d0,d1
500:            lsl.w   #8,d1
501:            bsr     one
502:            or.w    d1,d0
503:            and.l   #$ffff,d0
504:            bsr     jump
505:            bra     getp3
506:
507: getp6:     cmpi.b  #$15,d0             *ベースチャンネル
508:            bne     getp7
509:            bsr     one
510:            move.b  d0,base_ch-work(a6)
511:            bra     getp3
512:
513: getp7:     cmpi.b  #$1b,d0             *FM音色2
514:            bne     getp8
515:            moveq   #56,d0
516:            bsr     jump
517:            bra     getp3
518:
519: getp8:     cmpi.b  #$40,d0             *ADPCMコンフィギュレーション
520:            bne     getp10
521:            moveq   #19,d0
522:            bsr     jump
523:            bsr     one
524:            tst.b   d0
525:            bne     getp9
526:            moveq   #3,d0
527:            bsr     jump
528:            bra     getp3
529: getp9:     bsr     one
530:            tst.b   d0
531:            bne     getp9
532:            bra     getp3
533:
534: getp10:    cmpi.b  #$42,d0             *ベースカウンタ
535:            bne     getp11
536:            moveq   #5,d0
537:            bsr     jump
538:            bra     getp3
539: getp11:
540:            cmpi.b  #$61,d0             *PRINTコマンド
541:            bne     getp12
542:            bra     getp9
543:
544: getp12:    cmpi.b  #$62,d0             *MIDIダンプデータMDDファイル名
545:            bne     getp14
546:
547:            lea     mddfln_buf(pc),a1
548:            cmp.b   #-1,(a1)
549:            bne     getp9
550:            moveq   #9,d1
551: getp13:    bsr     one
552:            move.b  d0,(a1)+
553:            beq     getp3
554:            dbra    d1,getp13
555:            move.b  #0,-1(a1)
556:            bra     getp9
557:
558: getp14:    cmpi.b  #$63,d0             *ZPDデータファイル名
559:            bne     getp15
560:
561:            lea     zpdfln_buf(pc),a1
562:            cmp.b   #-1,(a1)
563:            bne     getp9
564:            moveq   #9,d1
565:            bra     getp13
566:
567: getp15:    cmpi.b  #$60,d0             *ADPCMコンフィギュレーションファイル名
568:            bne     getp16
569:
570:            lea     cnffln_buf(pc),a1
571:            cmp.b   #-1,(a1)
572:            bne     getp9
573:            moveq   #9,d1
574:            bra     getp13
575:
576: getp16:    cmpi.b  #$7f,d0             *コメント
577:            bne     getp17
578:
579:            lea     comment_buf(pc),a1
580:            cmp.b   #-1,(a1)
581:            bne     getp9
582:            moveq   #76,d1
583:            bra     getp13
584:
585: getp17:    cmpi.b  #$4a,d0             *波形メモリデータ
586:            bne     getp18
587:
588:            bsr     one
589:            move.l  d0,d1
590:            lsl.w   #8,d1
591:            bsr     one
592:            move.b  d0,d1
593:            moveq   #4,d0
594:            bsr     jump
595:            add.w   d1,d1
596:            move.l  d1,d0
597:            bsr     jump
598:            bra     getp3
599:
600: getp18:    cmpi.b  #$ff,d0             *共通コマンド部終了
601:            bne     getp3
602:
603: getp19:    bsr     one
604:            cmpi.b  #$ff,d0
605:            beq     getp19
606:            move.b  d0,d1
607:            lsl.w   #8,d1
608:            bsr     one
609:            move.b  d0,d1
610:            move.w  d1,num_of_trk-work(a6)  *トラック総数書き込み
611:            beq     getp_ret
612:
613: getp20:    moveq   #5,d0               *使用するデバイスをチェック
614:            bsr     jump
615:            bsr     one
616:            cmp.b   #8,d0
617:            bcs     getp22
618:            beq     getp21
619:
620:            cmpi.b  #25,d0
621:            bcc     getp21
622:                                        *MIDIのチャンネルが存在する
623:            move.b  #$ff,midi_on-work(a6)
624:            bra     getp23
625:
626: getp21:                                *ADPCMのチャンネルが存在する
627:            move.b  #$ff,pcm_on-work(a6)
628:            bra     getp23
629:
630: getp22:                                *FMのチャンネルが存在する
631:            move.b  #$ff,fm_on-work(a6)
632:
633: getp23:    subq.b  #1,d1
634:            bne     getp20
635:
636: getp_ret:
637:            rts
638:
639: ****************
640: one:                                   *1文字読み出し
641:            move.w  a3,-(sp)
642:            FNC     $1b
643:            addq.l  #2,sp
644:            rts
645:
646: ****************
647: jump:                                  *読み出し位置を前方へd0だけジャンプ
648:            move.w  #1,-(sp)
649:            move.l  d0,-(sp)
650:            move.w  a3,-(sp)
651:            FNC     $42
652:            addq.l  #8,sp
653:            rts
654:
655: ****************
656: valp_su:                               *数値10進書き込みサブ
657:
658:            lea     valp_v(pc),a1
659:            moveq   #0,d3
660:            ext.l   d1                  *今後のために65535までOKにした
661:            moveq   #0,d2               *d1-value
662:            move.l  #10000,d0
663: valp_l1:
664:            move.w  d1,d3
665:            divu    d0,d3
666:            move.b  d3,(a1,d2.w)
667:            mulu    d0,d3
668:            sub.w   d3,d1
669:            divu    #10,d0
670:            addq.w  #1,d2
671:            cmpi.b  #5,d2
672:            bne     valp_l1
673:
674:            moveq   #0,d0
675: valp_l2:
676:            move.b  (a1,d0.w),d1
677:            bne     valp_l3
678:            move.b  #$20,(a1,d0.w)
679:            addq.w  #1,d0
680:            cmpi.w  #4,d0
681:            beq     valp_l3
682:            bra     valp_l2
683:
684: valp_l3:
685:            add.b   #$30,(a1,d0.w)
686:            addq.w  #1,d0
687:            cmpi.w  #5,d0
688:            bne     valp_l3
689:            rts
690:
691: ********************************************
692: *          ワークエリア
693: ********************************************
694: work:
695: cur_dir:                    *カレントドライブ
696:            ds.b    1
697:            dc.b    ":¥"
698: path_buf:                   *カレントパス
699:            ds.b    256
700:            .even
701:
702: *******
703: fl_buf:                     *ファイル情報のためのバッファ
704:            dc.b    0
705:            ds.b    29
706: fl_name:                    *ファイルネームはここに取り込まれます
707:            ds.b    23
708:            ds.b    5           *ファイル名が溢れた時用
709:            .even
710:
711: *******
712: fln_txt:                    *ファイルを8文字で表示するための
713:            ds.b    8                   *バッファ
714:            dc.b    "  "
715: fln_txt_k:
716:            dc.b    ".xxx  ",0          *拡張子
717:
718:                        *各表示パラメータ用フラグ
719: flg_c:                      *コメントスイッチフラグ
720:            dc.b    0
721:
722: flg_z:                      *異常フォーマットフラグ
723:            dc.b    0
724:            .even
725:
726: ZmuSiC:
727:            ds.b    6           *ファイル先頭の「ZmuSic」を取り込む
728:
729:                                *各パラメータ取り込み用バッファ
730: base_ch:
731:            ds.b    1           *ベースチャンネルのモード（実は廃止されているらしい）
732:
733: comment_buf:                *コメント用バッファ
734:            ds.b    77
735:
736: zpdfln_buf:                 *指定のZPDファイルネーム
737:            ds.b    10
738:
739: cnffln_buf:                 *指定のADPCMコンフィグレーションファイルネーム
740:            ds.b    10
741:
742: mddfln_buf:                 *指定のMIDIダンプデータファイルネーム
743:            ds.b    10
744:
745: fm_on:                      *FM音源使用フラグ
746:            ds.b    1
747: pcm_on:                     *ADPCM使用フラグ
748:            ds.b    1
749: midi_on:                    *MIDI音源使用フラグ
750:            ds.b    1
751:            .even
752:
753: num_of_trk:                             *総トラック数
754:            ds.w    1
755:
756: ********************************
757: *          文字列データ
758: ********************************
759:
760: message1:
761:            dc.b    "スイッチの指定に誤りがあります。",13,10,0
762:
763: message2:
764:            dc.b    "指定ファイルがみつかりません。",13,10,0
765:
766: message3:
767:            dc.b    "このパスにZMDファイルはありません。",13,10,0
768:
769: message4:
770:            dc.b    $1b,"[36mThis Is Not ZMD File!",$1b,"[m",13,10,0
771:
772: message5:
773:            dc.b    "これはZMDファイルではありません。",13,10,0
774:
775: line_txt1:                                      *ラインに使います「-」が23個です。
776:            dc.b    $1b,"[37m-----------------------",$1b,"[m",0
777:
778: line_txt2:                                      *ラインに使います「=」が23個です。
779:            dc.b    $1b,"[37m=======================",$1b,"[m",0
780:
781: zmd_wild:
782:            dc.b    "*.ZMD",0
783:
784: txt_cr:
785:            dc.b    13,10,0
786:
787: para1_txt:
788:            dc.b    "・ZMD File Name  ",0
789:
790: para2_txt:
791:            dc.b    "・Comment",13,10,0
792:
793: para4_txt:
794:            dc.b    "・Track  ","・ZPD File Name  ","・CNF File Name  ","・MDD F
ile Name  "
795:            dc.b    "・Necessary Device  ",13,10,0
796:
797: valp_txt:                                       *数値表示に使っています
798:            dc.b    "  "
799:
800: valp_v:
801:            ds.b    5                   *5ケタ分
802:            dc.b    "  ",0
803:
804: fln_spc:                       *各種ファイルネームの指定がなかった場合の穴埋め表示
805:            dc.b    $1b,"[35m--------------  ",$1b,"[m",0
806:
807: fm_txt:
808:            dc.b    "FM ",0
809: pcm_txt:
810:            dc.b    "PCM ",0
811: midi_txt:
812:            dc.b    "MIDI ",0
813:
814:            .even
815:
816:            ds.b    128
817: usrsp:
```

▶「X-OVER・NIGHT」最終回ですか。高原さんごくろうさまでした。毎回とても面白かったです。
越智　亮(21)大阪府

［特集］
Z-MUSICシステム
ver. 2.0

アプリケーション作成のための

トラックワークの使い方

Horie Koutaro 堀江 孝太郎

強力なドライバとテクニックを駆使した演奏データ。これで，対応したアプリケーションがあるともっといい環境になります。ここではアプリケーション作成のためのZ-MUSICのトラックワークの使い方を探ります。

ついにZ-MUSICシステム ver.2.0が発表になりました。もともとZ-MUSICシステムは汎用の音楽ドライバとして位置付けられています。これは多くの意味を含んでいますが，プログラミングの際にさまざまなサービスをユーザーに提供するというのもそのうちのひとつに挙げられるでしょう。

しかし，実際にアプリケーションを作る段になってくると，ひととおりのシステムコールだけでは情報が足りないということもしばしば起こってきます。

たとえば，初期化されたグラフィック画面の任意の座標に1ドットが打たれたとき，グラフィックVRAMとパレットレジスタのアドレスおよび構造がわかっていれば，打たれたドットの座標，パレット番号，そのパレットのカラーコードという情報を得ることができます。では，音楽ドライバがFM音源で全音符のドを演奏しているとき，演奏されている音階や音長などの情報はどうすれば得られるのでしょうか？

FM音源レジスタのアドレスと構造はわかっているので，音階情報は一見調べられそうですが残念ながらレジスタが書き込み専用なので読み込みはできません。仮に読み書き両用の構造であったとしても，音長はドライバの固有情報なので調べようがありません。

音楽ドライバは演奏処理をしている最中に「ドを発音した」「全音符分音長をカウントしている」などの情報をワークに管理していますから，そのワークのアドレスおよび構造がわかっていれば外部からでも「全音符のドが演奏されている」という情報を得ることができます。

ZMUSIC.Xではワークのアドレス取得からワークの構造など，全情報が公開されています。もしも，システムの内部情報がまったく公開されていなければ，アプリケーションを作るときシステムの解析から行わなければなりません。システムの情報公開によってユーザーはアプリケーションの制作に専念することができます。システムを幅広く展開するためには当然のことです。

ここではビジュアルプレイヤーなどのアプリケーションを作るうえで，特に関係してくるワーク（演奏トラックワーク）について簡単に説明したいと思います。

●演奏トラックテーブル

ZMUSIC.Xは最大80本のトラックを確保でき，最大でそのうちの任意の32本を同時演奏することができます。トラックはメモリの確保(m_alloc)とトラックアサイン(m_assign)を行うことで演奏トラックとなります。

ファンクション＄3Aをコールすると演奏トラックを示したテーブル（演奏トラックテーブル）のアドレスを得ることができます。

演奏トラックテーブルは，たとえばトラック1，2，3，5を演奏するような曲データの場合，

　　$00,$01,$02,$04,$FF,……

（$FFはエンドコード。トラック番号－1である点に注意）

というようになっています。必ず詰まった状態で登録されていますので，＄FFを発見すれば以降は無効です。

●演奏トラックワーク

各トラックは演奏時，オフセット構造のワークエリアを使用しています。このワークエリアを演奏トラックワークと呼びます。演奏トラックワークは1トラックにつき256バイトあります。

ワークの一覧はlabel.sを見てのとおりです（表1参照)。

ファンクション＄3Cをコールすることによってトラック1の演奏トラックワークアドレスを得ることができ，トラックnの演奏トラックワークアドレスTWnは，

　　TW1＋(n－1)×256

で求めることができます（TW1はトラック1の演奏トラックワークアドレス)。

ZMUSIC.Xが常駐後，演奏トラックワークアドレスは絶対変動しません。アプリケーションでは最初にファンクション＄3Cをコールしたら，2度とコールしなくてもいいように保存しておくといいでしょう。

●ワーク参照上の注意点

ワークの中には，ひとつのワークアドレスに対して重複して2つ以上のワークが設定されているワークもあります。

例1）　同時に使用されることがない機能による重複ポルタメントとオートベンドは同時に使用できませんので，関連ワークは重複するオフセットアドレスに設定されています。

例2）　対象音源デバイスの相違による重複ワークアドレス＄A0は，MIDIではMIDI音色バリエーションの上位バイトが，FMではFM音源音色パラメータのオペレータ1のトータルレベル（アウトプットレベル）が設定されています。

●数値の対応

MMLやコマンドで与えたパラメータ値とワーク内部で表現される内部表現形式とが必ずしも一致しているわけではありません。たとえば音色番号はMMLパラメータでは1～200ですがワークp_pgmでは0～199となっています。

ワークエリアの書き換え

ワークエリアは情報を参照するだけでなく，外部から書き換えることでドライバの動作を制御することもできます。ミキサーのようなアプリケーションを作って演奏中の曲のあるトラックの音量を書き換えたり……といった使い方もできるわけです。

ただし，ワークエリアの不当な書き換えは演奏に支障をきたすだけでなく，ZMUSIC.Xの暴走を引き起こす可能性もありますので，書き換えには細心の注意が必要で

す。特殊な使用目的以外では，各ワークは読み出し専用と考え，書き込みはなるべくしないようにするほうがいいでしょう。

また，ワークの書き換えを正しく行っても演奏に影響が出ないことは多々あります。これはワークによってはドライバ処理の結果値でしかないものがあるからです。

音色番号
パンポット
MIDI音色バリエーション
MIDI汎用エフェクト1
(SC-55でいうリバーブ)
MIDI汎用エフェクト3
(SC-55でいうコーラス)

などは，書き換えによって演奏に影響が出ると面白そうなのですが残念ながら影響されません。

主なワークエリアの説明

では，1トラック256バイトある演奏トラックワークの中で基本となるものについて説明したいと思います。

●p_on_count(.W)　$00
●p_gate_time(.W)　$02

p_on_countとp_gate_timeはトラックを演奏していくうえで根本となるカウンタです。p_on_countはステップタイム，p_gate_timeはゲートタイムをキーオン時に書き込まれ，定められたテンポに従って－1されていきます。p_gate_timeが0になった時点でキーオフ処理が行われ，p_on_countが0になった時点で次の音符の演奏処理に移ります。

ステップタイムよりゲートタイムが大きいと当然キーオフ処理をされずに次の音符の演奏処理に移ってしまいます。この処理がZMUSIC.Xではタイ処理です。p_tie_flgというワークがありますがこれは前の音符にタイ/スラーの指定があったかどうかを表しています。現在の音符にタイ/スラーの指定があるかはp_on_countとp_gate_timeを比較してやればいいことになります。たとえば「q4c4d4&e4」が演奏されたとき，p_on_count, p_gate_time, p_tie_flgは次のように変化します。

1)　c4がキーオンする

p_on_countに48，p_gate_timeに24，p_tie_flgに0が書き込まれる

192(全音符のクロック数)÷4
＝48(4分音符のクロック数)
48(4分音符のクロック数)×4/8 (q4)
＝24(q4の4分音符のクロック数)
↓

表1　Z-MUSICのトラックワーク名とオフセット

ワーク名	オフセット
p_on_count(.W)	$00
p_gate_time(.W)	$02
p_data_pointer(.L)	$04
p_fo_spd(.B)	$08
p_ch(.B)	$09
p_not_empty(.B)	$0A
p_amod_step(.B)	$0B
p_mstep_tbl(.W)	$0C
p_wvpm_loop(.L)	$0C
p_wvpm_lpmd(.W)	$10
p_altp_flg(.B)	$12
p_fo_mode(.B)	$1C
p_pgm(.B)	$1D
p_pan(.B)	$1E
p_vol(.B)	$1F
p_mrvs_tbl(.B)	$20
p_wvpm_point(.L)	$20
p_wvpm_end(.L)	$24
p_sp_tie(.W)	$28
p_om(.B)	$28
p_sync(.B)	$29
p_af(.B)	$2A
p_se_mode(.B)	$2B
p_pmod_tbl(.W)	$2C
p_total(.L)	$3C
p_fo_lvl(.B)	$40
RESERVE(.B)	$41
p_note(.B)	$42
p_extra_ch(.B)	$4A
p_aftc_n(.B)	$4B
p_bend_rng_f(.W)	$4C
p_bend_rng_m(.W)	$4E
p_detune_f(.W)	$50
p_detune_m(.W)	$52
p_port_dly(.W)	$54
p_bend_dly(.W)	$56
p_port_work(.B)	$58
p_port_rvs(.B)	$59
p_port_work2(.W)	$5A
p_amod_tbl(.B)	$5C
p_arcc_tbl(.B)	$5C
p_arvs_tbl(.B)	$64
p_wvam_point(.L)	$64
p_wvam_end(.L)	$68
p_pmod_work4(.W)	$6C
p_port_flg(.W)	$6E
p_bend_flg(.W)	$70
p_aftc_tbl(.B)	$72
p_aftc_dly(.W)	$7A
p_aftc_work(.W)	$7C
p_astep_tbl(.B)	$7E
p_wvam_loop(.L)	$7E
p_wvam_lpmd(.W)	$82
p_alta_flg(.B)	$84
p_pmod_step2(.W)	$86
p_pmod_work(.W)	$88
p_pmod_work2(.W)	$8A
p_pmod_work3(.B)	$8C
p_pmod_n(.B)	$8D
p_sync_wk(.B)	$8E
p_rpt_last?(.B)	$8F
p_@b_range(.B)	$90
p_arcc(.B)	$91
p_pmod_flg(.W)	$92
p_pmod_sw(.B)	$94
p_amod_sw(.B)	$95
p_arcc_sw(.B)	$95
p_bend_sw(.B)	$96
p_aftc_flg(.B)	$97
p_md_flg(.B)	$98
p_waon_flg(.B)	$99
p_pmod_dly(.W)	$9A
p_amod_dly(.W)	$9C
p_arcc_dly(.W)	$9C
p_port_step(.W)	$9E
p_bank_msb(.B)	$A0
p_ol1(.B)	$A0
p_bank_lsb(.B)	$A1
p_ol2(.B)	$A1
p_effect1(.B)	$A2
p_ol3(.B)	$A2
p_effect3(.B)	$A3
p_ol4(.B)	$A3
p_d6_last(.B)	$A4
p_cf(.B)	$A4
p_amod_step2(.B)	$A5
p_pb_add(.B)	$A6
p_vset_flg(.B)	$A7
p_arcc_rst(.B)	$A8
p_arcc_def(.B)	$A9
p_coda_ptr(.L)	$AA
p_pointer(.L)	$AE
p_do_loop_ptr(.L)	$B2
p_pmod_work5(.W)	$B6
p_pmod_work6(.W)	$B8
p_amod_flg(.B)	$BA
p_arcc_flg(.B)	$BA
p_aftc_sw(.B)	$BB
p_dumper(.B)	$BC
p_tie_flg(.B)	$BD
p_pmod_dpt(.W)	$BE
p_seq_flag(.B)	$C0
p_do_loop_flag(.B)	$C1
p_pmod_spd(.W)	$C2
p_amod_spd(.W)	$C4
p_total_olp(.L)	$C6
p_pmod_step(.W)	$CA
p_tie_pmod(.B)	$CC
p_tie_bend(.B)	$CD
p_tie_amod(.B)	$CE
p_tie_arcc(.B)	$CE
p_tie_aftc(.B)	$CF
p_pan2(.B)	$D0
p_non_off(.B)	$D1
p_frq(.B)	$D2
p_velo(.B)	$D3
p_amod_work4(.W)	$D4
p_pmod_rvs(.B)	$D6
p_waon_dly(.B)	$D7
p_waon_work(.B)	$D8
p_waon_num(.B)	$D9
p_note_last(.B)	$D9
p_rpt_cnt(.B)	$DA
p_maker(.B)	$E2
p_device(.B)	$E3
p_module(.B)	$E4
p_last_aft(.B)	$E5
p_amod_work(.W)	$E6
p_arcc_work(.W)	$E6
p_arcc_work2(.B)	$E8
p_amod_work2(.B)	$E8
p_amod_work3(.B)	$E9
p_amod_work7(.B)	$EA
p_amod_n(.B)	$EB
p_arcc_n(.B)	$EB
p_arcc_work5(.W)	$EC
p_amod_work5(.W)	$EC
p_arcc_work6(.W)	$EE
p_amod_work6(.W)	$EE
p_pmod_wf(.B)	$F0
p_amod_dpt(.B)	$F1
p_amod_wf(.B)	$F2
p_dmp_n(.B)	$F3
p_pmod_omt(.B)	$F4
p_arcc_omt(.B)	$F5
p_amod_omt(.B)	$F5
p_pmod_mode(.B)	$F6
p_arcc_mode(.B)	$F7
p_pmod_chain(.B)	$F8
p_amod_chain(.B)	$F9
p_velo_dmy(.B)	$FA
p_waon_mark(.B)	$FB
p_marker(.W)	$FC
p_amod_rvs(.B)	$FE
p_ne_buff(.B)	$FF
p_user(.B)	$FF

ZWV.Xの画面

(24INT後, p_on_countが24, p_gate_timeが0になる)

↓

2)　キーオフする

↓

(24INT後, p_on_countが0になる)

↓

3)　d4がキーオンする

p_on_countに48, p_gate_timeに255, p_tie_flgに0が書き込まれる

↓

(48INT後, p_on_countが0になる, p_gate_timeは207)

↓

4)　e4がキーオンする。

p_tie_flgに$FFが書き込まれる

●**p_ch(.B)　$09**

●**p_extra_ch(.B)　$4a**

絶対チャンネルが登録されているp_chのワークの値域は0≦p_ch≦24です。PCM8独立チャンネルモードの場合はAD PCMの絶対チャンネルは26～32です。この絶対チャンネルが26～32のとき，p_chに8，p_extra_chに0～7の値が登録されています。p_chが8以外のとき，p_extra_chは不定となっています。

●**p_pan(.B)　$1e**

●**p_pan2(.B)　$d0**

パンポットのワークはp_panとp_pan2の2つがあります。パンポットはMMLにおいて左，右，中央の3段階設定のPコマンドと128段階設定の@Pコマンドの2つがありますが，どちらのコマンドを使用しても両方のワークに反映されます。アプリケーションでパンポット表示するときはFMとAD PCMをMIDIに合わせて多段階表示の場合はp_pan2のみ参照，MIDIをFM,AD PCMに合わせて3段階表示の場合はp_panのみ参照で問題ないでしょう。

●**p_note(.B)　$42～$49**

ノートナンバーが登録されているp_noteについて説明します。p_noteはマイナス値をエンドコードとして最大8個まで現在発音中の音階が書き込まれています。ワークサイズが8バイトのため8音発音中の場合はエンドコードはありません。たとえば，オクターブ4のCEG（ドミソ）を発音中の場合，

$3C,$40,$43,$FF,……

のようになっています（オクターブをn，音階をC（ド）を基準に0～11とすればノートナンバーKは，

K＝（n＋1）×12＋音階

で求められます。よって，オクターブ4のC，E，Gはそれぞれ$3C，$40，$43となります）。

ダンパーなどのディレイつき和音の場合は1音目のキーオン時に一括してすべてのノートナンバーがp_noteに登録され，p_waon_dly, p_waon_numに従って独立してキーオンされます。キーオフはノートナンバーの最上位ビットを立てたものか，あるいは$FFが書き込まれます。つまり上の例のキーオフ時には，

$BC,$C0,$C3,$FF,……

または，

$FF,$FF,$FF,$FF,……

となります。

●**p_seq_flag(.B)　$c0**

レベルメーターのようなものを作るときにはキーオン，キーオフなどを正確に検出する必要が出てきます。このような場合，p_seq_flagの第7ビットは非常に便利です。あらかじめアプリケーション側でビットクリアしておけば，ドライバ側でビットがセットされることで容易にキーオンを検出できます。また同様にして第6ビットによってキーオフの検出ができます。

簡易ワークビュアZWV.X

●**ZWV.Xについて**

ZWV.XはZMUSIC.Xの演奏トラックテーブル，演奏トラックワークの変化などリアルタイムに容易に見ることができるビュアです。

付加機能として実験的に特定のワークの書き換えをすることができます。比較的安全でエディット後の変化が聴覚的にも確認できるワークを選びました。

●**ZWV.Xのキー操作**

[ESC]

終了

[SPACE]

押している間，画面を保持します

[A]，[Z]

ベロシティ値を変化させます（FMでは直接変化しません）

p_veloを±1します

[S]，[X]

ボリューム値を変化させます(MIDIでは直接変化しません)

p_volを±1します

[D]，[C]

絶対チャンネルを変化させます（元の絶対チャンネルはキーオフ処理されないため音が残留してしまいます）

p_chを±1します

[F]，[V]

ディチューン値を変化させます

p_detune_fを±64，p_detune_mを±683します

[CTRL]

ゲートタイムを最大値にします(発音中の音がキーオンされなくなります)

p_gate_timeを－1にします

[ROLL UP]，[ROLL DOWN]

表示トラックを演奏トラックテーブルに従って変化させます

またマウスカーソルでワークアドレスを指すことによって，ワーク名を知ることができます。なお，掲載リスト量などの都合によりメッセージ表示量は不完全となっています（表示量の追加は簡単にできると思います）。

プログラムの補足・注意点

ZWV.XはPCM8モードのチェックと演奏開始のチェックをファンクション$50を利用して行っています。ファンクション$50をコールすることによってドライバの常駐状況などZMUSIC.Xの状態を示す情報テーブルを得ることができます。情報テーブルの内容は以下のとおりです。

コールパラメータ：なし

戻り値：　a0.l＝情報テーブルの格納アドレス

0(a0).b MIDIチャンネル有効か

≠0　有効

＝0　無効

1(a0).b 常駐ドライバのタイプ

＝$FFRS-MIDIバージョン

＝$00MIDI I/F(CZ-6BM1相当品)

＝$01POLYPHON-MIDIバージョン

2(a0).b 強制MIDIチャンネル有効モードか

≠0　YES

＝0　NO

3(a0).b AD PCMの発音にPCM8.Xを使用中か

≠0　YES

＝0　NO

4(a0).w PCM8モード

＜0　独立チャンネルモード

＞0　ポリモード

6(a0).b

第0ビット

＝1　ジュークボックス類が常駐している

＝0　常駐していない

第1ビット

＝1　演奏制御を割り込みで行うプログラムが常駐している(ZP /Dなど)

＝0　常駐していない

7(a0).b

＝$01　演奏待機状態

＝$FF　演奏開始

8(a0).b〜98(a0).b

直前に読んだファイルのファイル名(ZPD, MDD, CNFなど)

−$1(a0)〜−$8(a0)

エラーが発生した場合，そのエラーコードが，このエリアに格納されます。−$8(a0)がいちばん新しく発生したエラーで，−$1(a0)がいちばん古く発生したエラーです。古いものはどんどん切り捨てられていきます。アプリケーション側でこの8バイトを初期化($00に)しておけば，その時点から発生したエラーを知ることができます。

−$c(a0).l

Z-MUSICの割り込みが発生するたびにインクリメントされます。

ここをアプリケーション側でクリアしておけば，その時点からのステップカウントが求められます(Z-MUSICの割り込みは1ステップカウントごとに発生します)。

[D]，[C]キーによる絶対チャンネルの変更でFM, MIDIチャンネルからAD PCMチャンネルに変更するのは結構危険のようなので，AD PCMチャンネルには変更できないようにしてあります。またAD PCMチャンネルからの変更もできません。

キーオフされずに残った音はプレイヤーZP.Rでいったん演奏停止して消音するしかありません。注意してください。

リスト1

```
  1: *  ProjectZWV.
  2: *                              Programmed by あほみ
  3: *                              in 1993/11/16~
  4: *
  5:            INCLUDE iocscall.mac
  6:            INCLUDE doscall.mac
  7:
  8: *////////////////////////////////////////////////////////
  9:            .nlist
 10:
 11: BITSNS:             MACRO   n        *IOCS   _BITSNSと同等に機能する
 12:            move.b  $800+n.w,d0
 13:            ENDM
 14:
 15: B_LOCA:             MACRO   x,y
 16:            moveq   #x,d1
 17:            moveq   #y,d2
 18:            IOCS    _B_LOCATE
 19:            ENDM
 20:
 21: DISPLAY:   MACRO   mes,attr,x,y,len *メッセージ表示のマクロ
 22:            lea     mes(pc),a1
 23:            moveq   #attr,d1
 24:            moveq   #x,d2
 25:            moveq   #y,d3
 26:            moveq   #len-1,d4
 27:            IOCS    _B_PUTMES
 28:            ENDM
 29:
 30: *ZMSCCALL.MAC
 31: get_trk_tbl         equ     $3a
 32: get_play_work       equ     $3c
 33: picture_sync        equ     $43
 34: get_1st_comment     equ     $4e
 35: zm_status  equ      $50
 36:
 37: Z_MUSIC             MACRO   callname
 38:            moveq.l #callname,d1
 39:            trap    #3
 40:            ENDM
 41:
 42: *LABEL.S
 43: p_gate_time:        equ $02 *.w gate time
 44: p_ch:               equ $09 *.b アサインされているチャンネル
 45: p_not_empty:        equ $0a *.b トラックの生死(-1=dead/1=play end/0=al
ive)
 46: p_vol:              equ $1f *.b last volume(127~0)
 47: p_fo_lvl:  equ      $40 *.b 出力パーセンテージ(0-128)
 48: p_extra_ch:         equ $4a *.b 拡張チャンネル番号(PCM8 MODE専用0-7)
 49: p_detune_f:         equ $50 *.w デチューン(FM用の値)
 50: p_detune_m:         equ $52 *.w デチューン(MIDI用の値)
 51: p_md_flg:  equ      $98 *b0 @b:ベンド値をリセットすべきかどうか(MIDI専用 0=no
/1=yes)
 52: p_vset_flg:         equ $a7 *.b ボリュームリセットフラグ(FM)
 53: p_velo:             equ $d3 *.b velocity(0-127)
 54: p_user:             equ $ff *.b ユーザー汎用ワーク
 55:
 56: TX_PAL_ADDR         equ     $e82200
 57:
 58: *zwv.h
 59: LDX:                equ     8       *ダンプ表示位置(x)
 60: LDY:                equ     15      *ダンプ表示位置(y)
 61: LTX:                equ     8       *トラックNo.表示位置(x)
 62: LTY:                equ     13      *トラックNo.表示位置(y)
 63: LCX:                equ     0       *コメント表示位置(x)
 64: LCY:                equ     7       *コメント表示位置(y)
 65: LTTX:               equ     0       *演奏トラックテーブル表示位置(x)
 66: LTTY:               equ     10      *演奏トラックテーブル表示位置(y)
 67: LWNX:               equ     8       *ワークネーム表示位置(x)
 68: LWNY:               equ     31      *ワークネーム表示位置(y)
 69:
 70:            .list
 71: *////////////////////////////////////////////////////////////////////
/
 72: *
 73: *  プログラム開始・・・・・
 74: *
 75:            .text
 76:            .even
 77: zwv_start::
 78:    lea     stack_buf(pc),a0
 79:    move.l  sp,(a0)
 80:    move.l  a0,sp
 81:    lea     work_top(pc),a6
 82:
 83:    clr.l   a1
 84:    IOCS    _B_SUPER                * スーパーモード移行
 85:    move.l  d0,to_user_sp(a6)
 86:
 87:    bsr     zmsc_inst_chk
 88:
 89:    moveq   #$10,d1                 * 768×512モード
 90:    IOCS    _CRTMOD
 91:    move.w  #18,-(sp)               * カーソル非表示
 92:    DOS     _CONCTRL
 93:    addq.w  #2,sp
 94:
 95:    move.w  #%00100_00100_10000_0,TX_PAL_ADDR+2*1
 96:    move.w  #%11100_11100_11100_0,TX_PAL_ADDR+2*3
 97:
 98:    IOCS    _MS_INIT
 99:    move.l  #LDX*8*$10000+LDY*16,d1
100:    move.l  #((LDX+16*3-1)*8-1)*$10000+(LDY+16)*16-1,d2
101:    IOCS    _MS_LIMIT
102:    IOCS    _MS_CURST
103:    moveq   #0,d1
104:    IOCS    _SKEY_MOD
105:    IOCS    _MS_CURON
106:
107:    moveq   #-1,d2
108:    Z_MUSIC picture_sync            *同期モードオン
109:    moveq.l #1,d2
110:    Z_MUSIC get_play_work           * a0:seq_wk_tbl base addr.
111:    move.l  a0,zm_seq_wk_tbl(a6)
112:    Z_MUSIC zm_status               * a0:status_tbl base addr.
113:    st.b    7(a0)                   *とりあえず演奏開始フラグを立てておく
114:    move.l  a0,zm_status_tbl(a6)
115:    tst.b   3(a0)                   *PCM8モードか?
116:    sne.b   pcm8_flg(a6)
117:    Z_MUSIC get_trk_tbl             * a0:play_trk_tbl
118:    move.l  a0,zm_play_trk_tbl(a6)
119:    bsr     zwv_trk_no_init
120:
121:    bsr     zwv_scr_init
122:    st.b    no_chk_flg(a6)
123:    st.b    work_no(a6)
124: _main_lp:
125:    BITSNS  0
126:    btst    #1,d0                   *[ESC]
127:    bne     exec_fin
128: @@:        BITSNS  6
129:    btst    #5,d0                   *[SPACE]
130:    bne     @b
131: *
132:    move.l  zm_status_tbl(a6),a0
133:    tst.b   7(a0)                   *曲が切り替わってないか?
134:    beq     @f
135:    clr.b   7(a0)
136:    bsr     zwv_comm_wrt
137:    bsr     zwv_ptt_wrt
138:    bsr     zwv_trk_no_init
139:    st.b    no_chk_flg(a6)
140: @@:
141:    tst.b   no_chk_flg(a6)
142:    beq     @f
143:    bsr     zwv_optwrt
144: @@:
145:    bsr     zwv_dump
146:    clr.b   no_chk_flg(a6)
147:    bsr     zwv_key_job
148:    bsr     zwv_ms_job
149:    bra     _main_lp
150:
151: zwv_trk_no_init:
152:    move.l  zm_play_trk_tbl(a6),a0
153:    clr.w   zwv_trk_tbl_no(a6)
154:    moveq   #0,d0
155:    move.b  (a0),d0
156:    bpl     @f                *演奏トラックが少なくとも1つはあるか?
157:    moveq   #0,d0
158: @@:
159:    move.w  d0,zwv_trk_no(a6)
160:    rts
161:
162: zwv_ptt_wrt:                 *演奏トラックテーブルの表示
163:    B_LOCA  LTTX,LTTY+1
```

```
164:    move.l   zm_play_trk_tbl(a6),a0
165: @@:         move.b  (a0)+,d2
166:    bmi      @f                            *$FF?
167:    bsr      hex_put_1b
168:    moveq    #1,d1
169:    IOCS     _B_RIGHT
170:    bra      @b
171: @@:
172:    bsr      hex_put_1b
173:    moveq    #0,d1
174:    IOCS     _B_ERA_ST
175:    rts
176:
177: zwv_comm_wrt:                             *コメントの表示
178:    Z_MUSIC  get_1st_comment
179:    move.l   a0,a1
180:    moveq    #3,d1                         * d1:attr
181:    moveq    #LCX,d2                       * d2:x
182:    moveq    #LCY+1,d3                     * d3:y
183:    moveq    #96-1,d4                      * d4:length
184:    IOCS     _B_PUTMES
185:    rts
186:
187: zwv_optwrt:
188:    B_LOCA   LTX+11,LTY
189:    move.w   zwv_trk_no(a6),d2
190:    bsr      hex_put_1b
191:    bsr      zwv_devwrt
192:    rts
193:
194: zwv_devwrt:
195:    move.l   zm_seq_wk_tbl(a6),a4
196:    move.w   zwv_trk_no(a6),d0
197:    lsl.w    #8,d0
198:    adda.w   d0,a4                         * a4:trk_work_addr
199:    moveq    #0,d0
200:    move.b   p_ch(a4),d0
201:    cmpi.b   #8,d0
202:    bne      @f
203:    tst.b    pcm8_flg(a6)                  * PCM8モードではないか?
204:    beq      @f
205:    tst.b    p_extra_ch(a4)                * CH1か?
206:    beq      @f
207:    move.b   p_extra_ch(a4),d0
208:    addi.b   #24,d0
209: @@:
210:    mulu     #6,d0
211:    lea      dev_name_tbl(pc,d0.w),a1
212:    moveq    #3,d1
213:    moveq    #LDX+24,d2
214:    moveq    #LTY,d3
215:    moveq    #6-1,d4
216:    IOCS     _B_PUTMES
217:    rts
218:
219: dev_name_tbl:
220:    dc.b 'FMOPM1','FMOPM2','FMOPM3','FMOPM4','FMOPM5','FMOPM6','FMOP
M7','FMOPM8'
221:    dc.b 'ADPCM1','MIDI01','MIDI02','MIDI03','MIDI04','MIDI05','MIDI
06','MIDI07'
222:    dc.b 'MIDI08','MIDI09','MIDI10','MIDI11','MIDI12','MIDI13','MIDI
14','MIDI15'
223:    dc.b 'MIDI16','ADPCM2','ADPCM3','ADPCM4','ADPCM5','ADPCM6','ADPC
M7','ADPCM8'
224:
225: zwv_dump:
226:    move.l   zm_seq_wk_tbl(a6),a0
227:    move.w   zwv_trk_no(a6),d0
228:    lsl.w    #8,d0
229:    adda.w   d0,a0                         * a0:
230:    lea      preset_data(a6),a1            * a1:
231:    move.b   no_chk_flg(a6),d5
232:    moveq    #3,d1                         * d1:TX_PAL=3
233:    move.b   p_not_empty(a0),d0            * d0:p_not_empty(new)
234:    beq      @f                            *トラックが生きてるか?
235:    subq.w   #2,d1                         * d1:TX_PAL=1
236: @@:
237:    move.b   p_user(a0),d2                 * d2:p_not_empty(new)
238:    move.b   d0,p_user(a0)                 *汎用ワークにp_not_emptyを保存しておく
239:    cmp.b    d2,d0                         *p_not_emptyに変化があったか?
240:    beq      @f
241:    moveq    #-1,d5
242: @@:
243:    IOCS     _B_COLOR
244:    moveq    #LDX,d1                       * d1:
245:    moveq    #LDY,d2                       * d2:
246:    moveq    #16-1,d7
247: du_ly:      movem.l d1-d2,-(sp)
248:    moveq    #16-1,d6
249: du_lx:      movem.l d1-d2,-(sp)
250:    IOCS     _B_LOCATE
251:    moveq    #0,d2
252:    move.b   (a0)+,d2
253:    move.b   (a1),d1
254:    move.b   d2,(a1)+
255:    tst.b    d5
256:    bne      du_no_chk
257:    cmp.b    d1,d2                         * 値の変化はなかったか?
258:    beq      du_skip
259: du_no_chk:
260:    bsr      hex_put_1b
261: du_skip:
262:    movem.l  (sp)+,d1-d2
263:    addq.w   #3,d1
264:    dbra     d6,du_lx
265:    movem.l  (sp)+,d1-d2
266:    addq.w   #1,d2
267:    dbra     d7,du_ly
268:    moveq    #3,d1
269:    IOCS     _B_COLOR
270:    rts
271:
272: *  [in]    d2
273: hex_put_1b:
274:    move.l   d2,d1
275:    lsr.w    #4,d1
276:    bsr      hex_b_putc
277:    move.l   d2,d1
278:    andi.w   #15,d1
279:    bsr      hex_b_putc
280:    rts
281:
282: *  [in]    d1      0~15
283: hex_b_putc:
284:    move.b   hex_tbl(pc,d1.w),d1
285:    IOCS     _B_PUTC
286:    rts
287:
288: hex_tbl:    dc.b    '0123456789ABCDEF'
289:
290: zwv_key_job:
291:    IOCS     _B_KEYSNS
292:    tst.l    d0
293:    bne      @f
294:    rts
295: @@:
296:    lea      preset_data(a6),a5            * a5:preset_data_addr
297:    move.l   zm_seq_wk_tbl(a6),a4
298:    move.w   zwv_trk_no(a6),d0
299:    lsl.w    #8,d0
300:    adda.w   d0,a4                         * a4:trk_work_addr
301:    move.l   zm_play_trk_tbl(a6),a3        * a3:play_trk_tbl_addr
302:    IOCS     _B_KEYINP
303:    andi.w   #$ff00,d0
304:    cmpi.w   #$3800,d0                     *[ROLL UP]
305:    beq      kj_ROLL_UP
306:    cmpi.w   #$3900,d0                     *[ROLL DOWN]
307:    beq      kj_ROLL_DOWN
308:    cmpi.w   #$1e00,d0                     *[A]
309:    beq      kj_A
310:    cmpi.w   #$2a00,d0                     *[Z]
311:    beq      kj_Z
312:    cmpi.w   #$1f00,d0                     *[S]
313:    beq      kj_S
314:    cmpi.w   #$2b00,d0                     *[X]
315:    beq      kj_X
316:    cmpi.w   #$2000,d0                     *[D]
317:    beq      kj_D
318:    cmpi.w   #$2c00,d0                     *[C]
319:    beq      kj_C
320:    cmpi.w   #$2100,d0                     *[F]
321:    beq      kj_F
322:    cmpi.w   #$2d00,d0                     *[V]
323:    beq      kj_V
324:    cmpi.w   #$7100,d0                     *[CTRL]
325:    beq      kj_CTRL
326:    clr.b    p_vset_flg(a4)
327:    rts
328:
329: kj_ROLL_UP:
330:    move.w   zwv_trk_tbl_no(a6),d0
331:    tst.b    $00(a3,d0.w)
332:    bmi      kj_RUD_ret
333:    addq.w   #1,d0
334:    bra      kj_RUD
335:
336: kj_ROLL_DOWN:
337:    move.w   zwv_trk_tbl_no(a6),d0
338:    beq      kj_RUD_ret
339:    subq.w   #1,d0
340: kj_RUD:
341:    moveq    #0,d1
342:    move.b   $00(a3,d0.w),d1
343:    bmi      kj_RUD_ret
344:    move.w   d0,zwv_trk_tbl_no(a6)
345:    move.w   d1,zwv_trk_no(a6)
346:    st.b     no_chk_flg(a6)
347: kj_RUD_ret:
348:    rts
349:
350: kj_A:
351:    move.b   p_velo(a4),d0
352:    cmp.b    #127,d0
353:    beq      kj_AZ_ret
354:    addq.b   #1,d0
355:    bra      kj_AZ
356:
357: kj_Z:
358:    move.b   p_velo(a4),d0
359:    beq      kj_AZ_ret
360:    subq.b   #1,d0
361: kj_AZ:
362:    move.b   d0,p_velo(a4)
363: kj_AZ_ret:
364:    rts
365:
366: kj_S:
367:    move.b   p_vol(a4),d0
368:    cmp.b    #127,d0
369:    beq      kj_SX_ret
370:    addq.b   #1,d0
371:    bra      kj_SX
372:
373: kj_X:
374:    move.b   p_vol(a4),d0
375:    beq      kj_SX_ret
376:    subq.b   #1,d0
377: kj_SX:
378:    move.b   d0,p_vol(a4)
379:    st.b     p_vset_flg(a4)
380: kj_SX_ret:
381:    rts
382:
383: kj_D:
384:    move.b   p_ch(a4),d0
385:    cmpi.b   #8,d0              *ADPCMチャンネルか?
386:    beq      kj_DC_ret
387:    cmp.b    #24,d0
388:    beq      kj_DC_ret
389:    addq.b   #1,d0
390:    cmpi.b   #8,d0              *ADPCMチャンネルじゃないか?
391:    bne      @f
392:    addq.b   #1,d0
393: @@:
394:    bra      kj_DC
395:
396: kj_C:
397:    move.b   p_ch(a4),d0
398:    beq      kj_DC_ret
399:    cmpi.b   #8,d0              *ADPCMチャンネルか?
```

Z-MUSIC

▶「ストIIダッシュ」や「餓狼伝説２」がX68000でできるとは。ついにX68000にも格闘ゲームの波が本格的に押し寄せてきたようだ。　村上　健(25)福島県

```
400:     beq     kj_DC_ret
401:     subq.b  #1,d0
402:     cmpi.b  #8,d0           *ADPCMチャンネルじゃないか?
403:     bne     @f
404:     subq.b  #1,d0
405: @@:
406: kj_DC:
407:     move.b  d0,p_ch(a4)
408:     bsr     zwv_devwrt
409: kj_DC_ret:
410:     rts
411:
412: kj_F:
413:     move.w  p_detune_f(a4),d0
414:     addi.w  #64,d0          *FM音源:768÷12=64 (1半音)
415:     move.w  d0,p_detune_f(a4)
416:     move.w  p_detune_m(a4),d0
417:     addi.w  #683,d0         *MIDI音源:8192÷12≒683 (1半音)
418:     move.w  d0,p_detune_m(a4)
419:     bset.b  #0,p_md_flg(a4)
420:     rts
421:
422: kj_V:
423:     move.w  p_detune_f(a4),d0
424:     subi.w  #64,d0          *FM音源:768÷12=64 (1半音)
425:     move.w  d0,p_detune_f(a4)
426:     move.w  p_detune_m(a4),d0
427:     subi.w  #683,d0         *MIDI音源:8192÷12≒683 (1半音)
428:     move.w  d0,p_detune_m(a4)
429:     bset.b  #0,p_md_flg(a4)
430:     rts
431:
432: kj_CTRL:
433:     move.w  #$ffff,p_gate_time(a4)
434:     rts
435:
436: *//////////////////////////////////////////////////////////////////
437: zwv_ms_job:
438:     B_LOCA  LWNX+11,LWNY
439:     IOCS    _MS_CURGT
440:     subi.l  #LDX*8*$10000+LDY*16,d0
441:     move.l  d0,d2
442:     swap    d2
443:     ext.l   d2
444:     divu    #8*3,d2
445:     andi.w  #$00f0,d0
446:     or.w    d0,d2                   * d2:Work_No.
447:     move.b  work_no(a6),d1
448:     move.b  d2,work_no(a6)
449:     cmp.b   d1,d2
450:     beq     zwv_ms_job_ret
451:     add.w   d2,d2
452:     lea     LBASE(pc),a1
453:     move.w  $00(a1,d2.w),d1
454:     beq     @f
455:     adda.w  d1,a1
456:     IOCS    _B_PRINT
457: @@:
458:     moveq   #0,d1
459:     IOCS    _B_ERA_ST
460: zwv_ms_job_ret:
461:     rts
462:
463: LBASE:
464:     dcb.w   2,LF00-LBASE    *$00(.W)
465:     dcb.w   2,LF02-LBASE    *$02(.W)
466:     dcb.w   4,LF04-LBASE    *$04(.L)
467:     dc.w    0               *$08
468:     dc.w    LF09-LBASE      *$09(.B)
469:     dc.w    LF0a-LBASE      *$0a(.B)
470:     dcb.w   $12,0           *$0b~1c
471:     dc.w    LF1d-LBASE      *$1d(.B)
472:     dc.w    LF1e-LBASE      *$1e(.B)
473:     dc.w    LF1f-LBASE      *$1f(.B)
474:     dcb.w   $1c,0           *$20~3b
475:     dcb.w   4,LF3c-LBASE    *$3c(.L)
476:     dcb.w   2,0             *$40~41
477:     dcb.w   8,LF42-LBASE    *$42(.B)×8
478:     dc.w    LF4a-LBASE      *$4a(.B)
479:     dcb.w   5,0             *$4b~4f
480:     dcb.w   2,LF50-LBASE    *$50(.W)
481:     dcb.w   2,LF52-LBASE    *$52(.W)
482:     dcb.w   $45,0           *$54~98
483:     dc.w    LF99-LBASE      *$99(.B)
484:     dcb.w   $23,0           *$9a~bc
485:     dc.w    LFbd-LBASE      *$bd(.B)
486:     dcb.w   $12,0           *$be~cf
487:     dc.w    LFd0-LBASE      *$d0(.B)
488:     dcb.w   2,0             *$d1~d2
489:     dc.w    LFd3-LBASE      *$d3(.B)
490:     dcb.w   3,0             *$d4~d6
491:     dc.w    LFd7-LBASE      *$d7(.B)
492:     dcb.w   $27,0           *$d8~fe
493:     dc.w    LFff-LBASE      *$ff(.B)
494:
495: LF00:       dc.b '$00 p_on_count(.W)/ステップカウント',0    *$00
496: LF02:       dc.b '$02 p_gate_time(.W)/ゲートカウント',0     *$02
497: LF04:       dc.b '$04 p_data_pointer(.L)/データアドレス',0 *$04
498: LF09:       dc.b '$09 p_ch(.B)/絶対チャンネル',0            *$09
499: LF0a:       dc.b '$0a p_not_empty(.B)/トラック状態',0       *$0a
500: LF1d:       dc.b '$1d p_pgm(.B)/音色番号',0                 *$1d
501: LF1e:       dc.b '$1e p_pan(.B)/パンポット(FM)',0           *$1e
502: LF1f:       dc.b '$1f p_vol(.B)/ボリューム',0               *$1f
503: LF3c:       dc.b '$3c p_total(.L)/総カウント数',0           *$3c
504: LF42:       dc.b '$42~$49 p_note(.B)/キーコード',0   *$42~$49
505: LF4a:       dc.b '$4a p_extra_ch(.B)/拡張絶対チャンネル',0 *$4a
506: LF50:       dc.b '$50 p_detune_f(.W)/デチューン(FM)',0      *$50
507: LF52:       dc.b '$52 p_detune_m(.W)/デチューン(MIDI)',0    *$52
508: LF99:       dc.b '$99 p_waon_flg(.B)/和音フラグ',0          *$99
509: LFbd:       dc.b '$bd p_tie_flg(.B)/タイフラグ',0           *$bd
510: LFd0:       dc.b '$d0 p_pan2(.B)/パンポット(MIDI)',0        *$d0
511: LFd3:       dc.b '$d3 p_velo(.B)/ベロシティ',0              *$d3
512: LFd7:       dc.b '$d7 p_waon_dly(.B)/和音ディレイ値',0      *$d7
513: LFff:       dc.b '$ff p_user(.B)/ユーザー汎用ワーク',0      *$ff
514:     .even
515:
516: *//////////////////////////////////////////////////////////////////
517: *
518: *  プログラム終了……
519: *
520: exec_fin:
521:     moveq   #0,d2
522:     Z_MUSIC picture_sync                *同期モードオフ
523:
524:     IOCS    _MS_CUROF
525:     IOCS    _MS_INIT
526:     moveq   #-1,d1
527:     IOCS    _SKEY_MOD
528:     moveq   #$10,d1
529:     IOCS    _CRTMOD
530:
531:     moveq   #3,d1
532:     IOCS    _B_COLOR
533:     clr.w   -(sp)
534:     move.w  #14,-(sp)                   * ファンクションキー表示
535:     DOS     _CONCTRL
536:     move.w  #31,-(sp)
537:     clr.w   -(sp)
538:     move.w  #15,-(sp)                   * スクロール範囲設定
539:     DOS     _CONCTRL
540:     move.w  #17,-(sp)                   * カーソル表示
541:     DOS     _CONCTRL
542:     move.w  #$0000,-(sp)                * キーバッファ初期化
543:     DOS     _KFLUSH
544:     lea     14(sp),sp
545: exec_fin02:
546:     move.l  to_user_sp(a6),a1
547:     IOCS    _B_SUPER                    * ユーザーモード移行
548: EXIT:
549:     movea.l stack_buf(pc),sp
550:     DOS     _EXIT
551:
552: *//////////////////////////////////////////////////////////////////
553: zmsc_inst_chk:
554:     move.l  $8c.w,a0
555:     subq.w  #8,a0
556:     cmpi.l  #'ZmuS',(a0)+
557:     bne     zmsc_inst_err
558:     cmpi.w  #'iC',(a0)+
559:     bne     zmsc_inst_err
560:     move.w  (a0)+,d0
561:     andi.w  #$ff0f,d0
562:     cmpi.w  #$1500,d0                   *バージョンコード=1.50
563:     bmi     zmsc_ver_err
564:     rts
565: zmsc_inst_err:
566:     pea     ZMSC_INST_ERRMES01(pc)
567:     DOS     _PRINT
568:     bra     exec_fin02
569: zmsc_ver_err:
570:     pea     ZMSC_INST_ERRMES02(pc)
571:     DOS     _PRINT
572:     bra     exec_fin02
573: ZMSC_INST_ERRMES02:
574:     dc.b    'バージョン1.50以降の'
575: ZMSC_INST_ERRMES01:
576:     dc.b    'Z-MUSICが常駐していません',13,10,0
577:     .even
578: *//////////////////////////////////////////////////////////////////
579: zwv_scr_init:
580:     DISPLAY STR_00,2,LDX,LDY-1,3*16
581:     lea     STR_01(pc),a1
582:     lea     hex_tbl+1(pc),a2
583:     moveq   #2,d1                           * d1:attr
584:     moveq   #LDX-4,d2                       * d2:x
585:     moveq   #LDY,d3                         * d3:y
586:     moveq   #3,d4                           * d4:length
587:     moveq   #16-1,d7
588: @@:         movem.l d1-d4/a1,-(sp)
589:     IOCS    _B_PUTMES
590:     movem.l (sp)+,d1-d4/a1
591:     move.b  (a2)+,1(a1)
592:     addq.w  #1,d3
593:     dbra    d7,@b
594:     DISPLAY STR_02,3+4,LTX+00,LTY,11
595:     DISPLAY STR_03,3+4,LTX+16,LTY,6
596:     DISPLAY STR_04,3+4,LWNX,LWNY,9
597:     DISPLAY STR_10,3+4,LCX,LCY,11
598:     DISPLAY STR_20,3+4,LTTX,LTTY,21
599:     rts
600:
601: *             0         1         2         3         4         5
    6         7
602: *             12345678901234567890123456789012345678901234567890123
45678901234567890
603: STR_00: dc.b '+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F',0
604: STR_01: dc.b '$00',0
605: STR_02: dc.b 'TRACK NO. $',0
606: STR_03: dc.b 'DEVICE',0
607: STR_04: dc.b 'WORK NAME',0
608: STR_10: dc.b '1ST COMMENT',0
609: STR_20: dc.b 'PLAY TRACK TABLE',0
610:
611: *//////////////////////////////////////////////////////////////////
612:                     .offset 0
613: to_user_sp:         ds.l    1     *
614:
615: zwv_trk_tbl_no:     ds.w    1     * 現在表示しているトラックテーブルNo.(0~
79)
616: zwv_trk_no:         ds.w    1     * 現在表示しているトラックNo.(0~31)
617: zm_seq_wk_tbl:      ds.l    1     * func_$3cで得た演奏ﾜｰｸ先頭ｱﾄﾞﾚｽ
618: zm_status_tbl:      ds.l    1     * func_$50で得たｽﾃｰﾀｽﾃｰﾌﾞﾙｱﾄﾞﾚｽ
619: zm_play_trk_tbl:    ds.l    1     * func_$3aで得たplay_trk_tblｱﾄﾞﾚｽ
620: work_no:            ds.b    1     *
621: preset_data:        ds.b    256   * ﾌﾟﾘｾｯﾄ･ﾃﾞｰﾀ
622: no_chk_flg:         ds.b    1
623: pcm8_flg:           ds.b    1
624: work_size:
625:                     .bss
626:                     .even
627: work_top:           ds.b    work_size
628: *//////////////////////////////////////////////////////////////////
629:                     .stack
630:                     .even
631:                     ds.l    1024
632: stack_buf:
633:                     ds.l    1
634: end_of_prog:
635:                     .end    zwv_start
```

▶ああ，刻一刻と時が過ぎ去っていく……。　　藤田　康一(22)静岡県

新ZVT.Xによる音の加工

畳み込めば君もサウンドクリエイター

Nishikawa Zenji 西川 善司

今回のバージョンアップで特に強化された点としてAD PCMまわりが挙げられます。ここでは特にわかりにくいインパルスデータの畳み込みについて解説します。

Z-MUSICの標準ツールであるZVT.X(あるいはZPLK.R)に搭載されている「畳み込み(CONVOLUTION)」とはいったいなんなのでしょうか。専門的な言葉でいうと「インパルス応答から任意の有界な励振(Bounded Excitation)に対する応答を算出するための計算法」です。これだけではなんだかよくわかんないですね。でも大丈夫。この記事は基礎知識をいっさい持たない人にでも理解できるように進めていきますから安心して読み進んでください。

畳み込みでできること

畳み込み演算はデータの暗号化,通信手段のための符号化などさまざまな分野で応用されている演算です。電子/情報系などの理系大学生ならばなんらかのかたちで耳にしているかもしれません。

音声分野に限っていえばZVT.Xでも行っているような2つの音声の合成処理手段としてよく用いられます。たとえば美しいバイオリンの音色とあなたのオンチなダミ声(失礼)を畳み込みで合成したとすると,あなたの声にバイオリンの美しい響きが乗り天使のささやきのような幻想的な音になります。

まったく無関係な2つの音を畳み込むと想像を絶する奇妙な音が生まれることがあります。畳み込みの用途の楽しさはここにあります。

奇妙な音はゲームなどの効果音や意表を突いたSEパーカッションには最適です。このようなことから畳み込みは「音の結婚」と呼ばれることもあります。

最近ステレオコンポによく搭載されているDSP機能とかデジタルエフェクタのリバーブなどにも畳み込みが応用されています。この場合はステレオやエフェクタの中のメモリに格納されている残響特性インパルスデータというものと,入力の音声信号を畳み込んで,この演算結果と入力の原音をミックスして出力しています。

インパルスデータは,ホールの中できわめて減衰速度の速い音を鳴らしてこれを採集したり,逆畳み込み演算を行って採集したりします。

このインパルスデータがあればそのホールの中で任意の音を演奏した結果がシミュレートできるわけです。たとえば東京ドームのインパルスデータがあったとして,このデータとあなたのダミ声(失礼)を畳み込めば東京ドームの中であなたがダミ声で叫んだ音が生まれるのです。

ただし,インパルスデータは各エフェクタメーカーの財産なので一般ユーザーが入手することは困難です。しかしちょっとした工夫でインパルスデータを合成することができます。この辺については後述します。

畳み込みはX68000などのパソコンで処理するとけっこうな時間を食ってしまいます。そんな時間のかかる演算をステレオやエフェクタはどのように処理しているのでしょうか。普通に処理したのではエフェクタに入力した音が残響効果を加えられて出てくるまでタイムラグが多すぎて実用にならないでしょう。

本来ならば時間のかかる畳み込み演算を高速にリアルタイムに並列処理してしまうのがいわゆるDSPチップなのです。X68000にはDSPチップは搭載されていませんので残念ながら畳み込み演算は少々時間のかかる処理になってしまっています。ご了承ください。

畳み込みのアルゴリズム

2組のN点の有限離散時間信号x(n)とh(n)(n=0,1,2,…,N−1)の畳み込みy(n)は,

$$y(n)=\sum_{i=0}^{N-1}x(i)h(n-i)$$

で表されます。この式だとなんだか難しいような感じがしますが実はこれは小学生が普段計算問題を解くときに使っている手段とそうたいして変わりません。

それでは1ステップずつ噛み砕いて解説していきましょう。

2つの音声データ同士の畳み込みにしろ,インパルスデータとの畳み込みにしろ,演算はPCMデータレベル(AD PCMではなくリニアPCMです)として行われます(インパルスデータには残響特性と偉そうな肩書きがありますが結局のところPCMデータです)。

この2つのPCMデータをX,Hとし,求めたい結果をYとします。PCMデータXのデータの長さをmとし,配列x(0)~x(m−1)にPCMデータXの各PCM値が格納されているものとします。

同様にPCMデータHのほうも長さをn,配列h(0)~h(n−1)にPCMデータHの各PCM値が格納されているものとします。結果Yを格納する配列y()を用意しておきこれを0で初期化しておきます。配列y()はy(0)~y((m+n−1)−1)まで取っておきます。

乗算の結果,演算結果のビット長は最大でPCMデータXのビット長とPCMデータHのビット長の和になるので,演算結果の格納先である配列y()のビット長は必ずその大きさのビット長に設定します。たとえばPCMデータXとPCMデータHのビット長がともに16だったら配列y()のビット数は32とするわけです(ここまで図1)。

ここから畳み込みの計算を始めます。x(0)とh(0)~h(n−1)を乗算します。結果であるx(0)h(0),x(0)h(1),……x(0)h(n−1)をy(0),y(1),……,y(n−1)へ格納します(ここまで図2)。

同様に今度はx(1)とh(0)~h(n−1)を乗算し,結果のx(1)h(0),x(1)h(1),……x(1)h(n−1)を今度は先ほどより1ずらしたy

(1)よりすでに格納してある値に加算格納していきます。つまりy(1), y(2), ……, y(n)へ加算格納します(ここまで図3)。

このようにしてx(2)からx(m−1)までも同様にしてh(0)〜h(n−1)と乗算し，結果をy( )へ格納先を1ずつずらして加算格納していきます。演算に使用されたxの要素番号と同じ番号のyから格納していくという感じです。たとえばx(15)ならばx(15)h(0)をy(15)へ, x(15)h(1)をy(16)へ, x(15)h(n−1)をy(15+n−1)へ……となります。

図1 準備

図2 1回目の乗算

図3 2回目の乗算

ここで簡単な例を使って実際に計算していきましょう。PCMデータXをx(0)=3, x(1)=5, x(2)=−7とし，PCMデータHをh(0)=12, h(1)=−4とした場合はどうなるでしょうか。答えは図4に示しておきます。

いまの例題をやって気づいた人もいるでしょうか。計算のアルゴリズムが掛け算の筆算に非常に似ているのです。先ほど「小学生の計算問題のようなもの」といったのはそのためです。繰り上がりがないという違いはありますが，ほぼそのものといった感じです。格納先y( )の大きさが(m+n−1)という理由もこれで納得できたでしょう。筆算の掛け算において桁数がm, nのものは繰り上がりがなければ答えの桁数は必ず(m+n−1)になるからです。

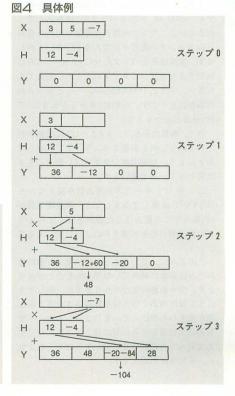
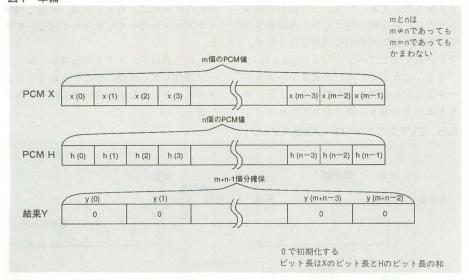
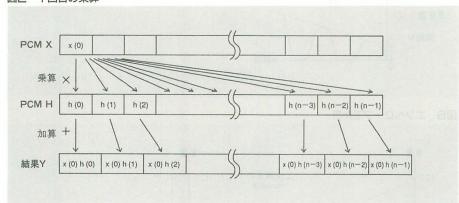
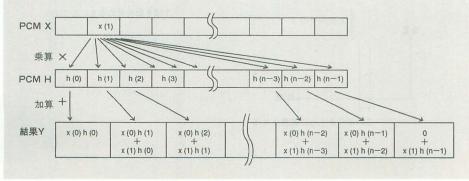
さて，ここまでで必要な処理のほとんどが完了していますが，まだ完全ではありません。畳み込んだ2つのPCMが16ビットレンジだった場合，乗算によって2倍の32ビットレンジに拡張されてしまっています。つまり，元のダイナミックレンジの音声信号に戻すには32ビット→16ビット変換が必要になります。

これを実行するには演算結果のYを2^{16}(=65536)で割ればいいことになります。具体的には右へ16ビットシフトしたり(ASR.L #16,Dn)，上位ワードと下位ワードを逆転させたり(SWAP Dn)などの方法があり，この処理を施してやっとYが再生可能なPCMデータになります。

インパルスデータを作ろう

初めにいったようにホールやスタジオなどの残響特性データであるインパルスデータは普通入手することは困難です。友人に音響研の大学院生がいるとか，エフェクタのROMデータを吸い出せる人と友達だとかかなり特殊な人脈がないと入手できないでしょう。

それならば作ってしまいましょう。とはいってもまぁ，結論からいえばあまり素晴

図4 具体例

らしいものではないですが，それなりのものが作れます。

それは……なんとノイズから作り出せます。まず，FMラジオのチューニング不良時のノイズやテレビの砂嵐でもなんでもいいですからホワイトノイズをサンプリングしましょう。

ホワイトノイズとはサーとかザーとか一様な音程感のあるノイズです。ギュイーーンとかボボボーなど揺らぎの激しいものでもかまいませんが，あまりに特異なものだと残響用のインパルスデータにはなりません。もっともトリッキーな音を作りたいというのであればあえてそういったノイズをソースに選択するのも悪くないかもしれません。

さて，ノイズをZVT.Xなどでサンプリングしたら，適当な大きさに切り出してください。あまり長いと演算に時間がかかりますので初めは短いほうがいいでしょう。切り出しには波形カーソルを使って［C］コマンドで切り出します。

そのノイズ波形を，音量がだんだんと下がっていくようなエンベロープに加工してやります。エンベロープ形状の変更はエフェクトコマンドのメニュー9番([E] キーを押してから [9] キー)で行います。

ZVT.Xではエンベロープの変更パターンを2つ持っています。変更パターンとその仕組みは図5のようになっています。ここでは，ノイズの音量を徐々に減衰するようなエンベロープにしたいので変更パターンは2番でその最終レベルは0にします。

普通にエンベロープ変更を実行すると，最大音量から線形的に最終レベルで設定した音量へ減衰していく単純なモデルになります。よりリアルなものを目指したいならば頭から減衰させずに頭の部分は原音量のまま少し残しておいたり，一度線形に減衰させたところを部分的にもう一度減衰させて疑似的に指数関数的な減衰を作り出す，など各自工夫してみてください(図6)。

さて，こうしてできた合成インパルスデータとなにかを畳み込んでみましょう。やり方は左下の囲みと同じ要領で行ってください。せっかく残響がつくかつかないかを試すのですから，実験に使用する音声はなるべく残響感の少ないドライな音にしましょう。

さて，結果はいかがだったでしょうか。音の輪郭がハッキリせず残響音のみのような結果になってしまった場合は，この結果と原音とを適当な音量バランスにてミックス(ZVT.Xの［+］キーコマンド)してみてください。そうすれば，エフェクタのリバーブを通したような音になるはずです。

大本は単なるノイズだったはずなのに，ちょっとエンベロープを変えるだけで，畳み込んでみれば，あら不思議，立派な残響インパルスになってしまったではないですか。

というわけで，ノイズの種類やエンベロープの作り方で残響の感じが変わってきますのでいろいろな組み合わせで試してみましょう。

実際に畳み込んでみよう

Z-MUSIC ver.2.0付属のPCMディスクの効果音ディスクライブラリからMTLCRSH.PCM(金属同士がぶつかる音)とSCRM_ML.PCM(男性の悲鳴)を持ってきてこの2つを畳み込んでみましょう。

まず，ZVT.Xを起動してMTLCRSH.PCMをバンクA，SCRM_ML.PCMをバンクBに読み込んでください。ここで2つの波形が画面に出たら畳み込みコマンドの実行キーである［Z］キーを押します。演算結果をバンクA，バンクBのどちらに格納するかを聞いてきますのでこれに答えてください。すると「PROCESSING……」と表示され計算が始まります。X68030(キャッシュオン)で約5分ほどかかります。

さて，演算結果がバンクAまたはBに格納されているのでこれを演奏コマンド［P］で演奏してみましょう。悲鳴の声が金属っぽい響きに生まれ変わっていると思います。

ここで［F］キーで再生周波数を変えながら［P］キーで演奏してみましょう。再生周波数が10.4kHzでこの畳み込んだ音声を聞くとちょうど怪獣映画の怪獣の鳴き声のように聞こえるでしょう。

このように，2つのまったく無関係な音の畳み込みは非常に興味深い結果を生むことがあります。少々演算時間がかかりますが，暇なときにでも無作為の2つの音を畳み込ませてどんな音が生まれるか試してみましょう。また，一度畳み込んだ音をさらに別の音と畳み込んだりするといっそう奇妙な音に変化するかもしれません。可能性は無限大です。

図5 ZVT.Xのエンベロープ変更パターン

図6 エンベロープ工夫例

フーリエ変換を使って高速に畳み込み

畳み込みはパソコンでは非常に時間のかかる演算処理です。現在ZVT.Xでは定義どおりの計算式，

$$y(n) = \sum_{i=0}^{N-1} x(i)h(n-i)$$

で演算を行っています。こちらの方法は整数領域ですべての演算を終了できるので手軽なのですが，長さNの2つのデータを畳み込んだときは計算量がN^2となってしまいます。つまり，データが長くなればなるほど計算量が多くなり終了まで時間がかかるということです。

しかもこの「計算」が加算減算だけならばまだしも，MPU68000の演算命令のなかでもかなり遅い部類の命令である「乗算」も使用しますからなおさらです。

この時間の非常にかかってしまう畳み込みを高速に行うアルゴリズムが考え出されているので紹介します。

高速フーリエ変換(FFT)を用いる方法です。入力のデータであるx( )とh( )にFFTを施し，その結果の各項同士の積を求め，これを逆FFTにかけて答えを求めるというものです。FFTといえば本誌1991年12月号で石上達也氏がやった「冬の夜長のスペクトル解析」というスカしたタイトルで解説されているように，スペクトル解析などに使われるのが一般的ですが，とてつもない桁数同士の乗算(円周率の計算など)をより少ない計算量で高速に行うためなどにも使われたりします。今回は後者の利用法でFFTを使うわけです。

FFTについての詳しい解説はデジタル信号処理や情報処理の専門書のほうを参照してください。誌面の都合がありますのでFFTはここでは単なるデータの変換手段の道具として用いてしまいます。

前の例同様のPCMデータXとPCMデータHの2つがあったとします。これをFFTにかけるわけですが，そのデータ長がFFTプログラムに対応した長さになっていないと正しい結果が得られません。

たとえば一般的なC言語のライブラリなどのFFTは2のべき乗の長さを持つデータに対してのFFTです。FFTを施したいデータの長さがこの2のべき乗に一致していないとダメということです。具体的な例を取り上げた専門書には(手元にはなかったのですが)3，4，8のべき乗のデータ長に対してのFFTプログラムが載っているかと思います。

FFTについての理解が深まれば自力で2のべき乗のFFTプログラムから別の数のべき乗のFFTプログラムをモディファイすることもできるでしょう。

まあ，実際のPCMデータが2のべき乗の長さと一致するということはまずないと思います。そういう場合は2のべき乗の長さに達するところまで0を付け足して，無理矢理2のべき乗のデータ長にしてください。

たとえば250バイトのデータがあったとき，これにいちばん近い2のべき乗の数は256です。ですから，このデータの後ろに0を6個足してやり，無理矢理256バイトのデータとしてやるわけです。もし，729バイトのデータがあったときはどうしたらいいでしょうか。

これに近い2のべき乗といえば1024です。1024−729=295個で295の0を後ろにくっつけて無理矢理1024バイトのデータとするわけですが，もし，3のべき乗対応のFFTプログラムを用意しておけば，ちょうど729は3^6ですから，0で帳尻を合わせる必要がないわけです。

0で帳尻を合わせたデータのFFTは付け足した0の部分に対しての余計な計算が入りますから，演算時間の増加につながります。そういうわけで，なるべく0の付け足しによる帳尻合わせが少ない手段を選択すべきなのです。

2つの入力データに0を付け足して2つとも同じサイズにしてください。たとえば2のべき乗のFFTを行う場合に，それぞれの長さが250バイトと280バイトだとします。0をそれぞれにいくつか補うことによって作り出せる共通の2のべき乗の数というと$2^9=512$です。250バイトのほうは6個補って256にできますが，280のほうは256を超えていますので，いちばん近いのは512です。そこで長いほうにあわせますから250バイトのほうは262個の0を補って512バイトへ，280バイトのほうは232個の0を補って512バイトにするわけです。

先ほどもいったように0を補う数が増えれば増えるほど，余分な計算が増えるわけですから，2つの入力データの長さが著しく違う場合は，FFTを用いないで計算したほうが早く結果が出る……ということもありうるわけです。

ここまでの下準備が図7です。

2つの入力に対してFFTを行うと，実数部と虚数部に別れた結果が出てきます。FFTとは離散フーリエ変換(DFT)の高速版です。N点の離散フーリエ変換とは，

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j(2\pi/N)nk}$$

$$(k=0,\ 1,\ \cdots,\ N-1)$$

ですから，ほら，虚数が出てくるのは当たり前ですね$(j=\sqrt{-1})$。さて，FFT前にそれぞれの入力の長さを複数個の0を補って帳尻を合わせたため，FFTの結果も，それぞれの実数部と虚数部の個数は一致しています。FFT後は，

$a+jb$　(a，bは実数)

というかたちになっています。ここで，対応した1個1個の項同士を複素乗算してやります。

複素乗算とは，たとえば，

$(a+jb)\times(c+jd)$　(a, b, c, dは実数)

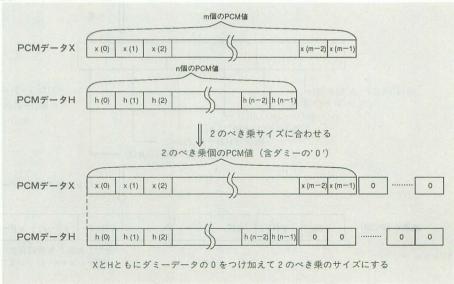
は，

$ac+jad+jbc-bd$

つまり，

$(ac-bd)+j(ad+bc)$

図7 FFT前の下準備

XとHともにダミーデータの0をつけ加えて2のべき乗のサイズにする

になります。

2つの16ビットのPCMをFFTし，結果が実数になり，さらに実数同士の乗算を，ある固定ビット数の数値表現にするということは当然桁落ちが発生するわけで，誤差となります。

そういうわけで，本来ならば，この誤差の問題に真剣に取り組まなければならないのですが，我々は数値計算をしているのではなくて音声データの畳み込みを行っています。ですから，音になって聞こえさえすれば多少の誤差は無視できるものとして考えないことにします。

数億桁の円周率の計算をする場合は，この誤差により演算結果のある桁以上は間違いだらけ……なんていう悲惨なケースもあります。この誤差により，FFTと非FFTによる畳み込みのそれぞれの結果が数値レベルでかなり違ってくるという状況もありえます。まぁ，でも，出てくる音にそう違いはないはずです。

さて，複素乗算を行って，結果も複素数ですから，実数部と虚数部に分けて格納します。これを逆フーリエ変換してすべての作業の終了です。離散逆フーリエ変換は，

$$x(n)=(1/N)\sum_{k=0}^{N-1}X(k)\cdot e^{j(2\pi/N)nk}$$

(k=0, 1, …, N−1)

です。FFTのプログラムを少し変えるだけで済みます。

(1/N)というのが目について「除算は遅いんだぞ!」と声を張り上げそうな人がいるかもしれませんが，Nは下準備の段階で2のべき乗の値ですから，結局除算というのは右シフト，もしくは指数部への減算で済むことになります。3のべき乗のFFTなどを用いた場合は少々話が違ってきますが。

逆FFTが完了すれば，畳み込み演算の完了したPCMデータが完成します。結果の大きさは2のべき乗サイズですが，有効な値は整数積算の方法と同じy(0)からy((m+n−1)−1)まででです。

ここまでが図8です。

結局，FFTを用いて畳み込みを行うとNが2のべき乗の場合，$3N\cdot\log_2 N+N$回の複素乗算で計算ができるようになります。

ただしFFTはいつでも本当に速いかというとそうとは限りません。前述の2つの入力の長さが違いすぎる場合はFFTの際に補う0の数が増え，結果的に余分な計算を強いられ演算量が余計にかさむ場合があります。

整数演算よりも格段に遅い浮動小数点の演算が不可欠になりますから，短いデータだと，FFTを用いるより通常の定義どおりの整数積算の方法によって行ったほうが早く済むという場合もあります。

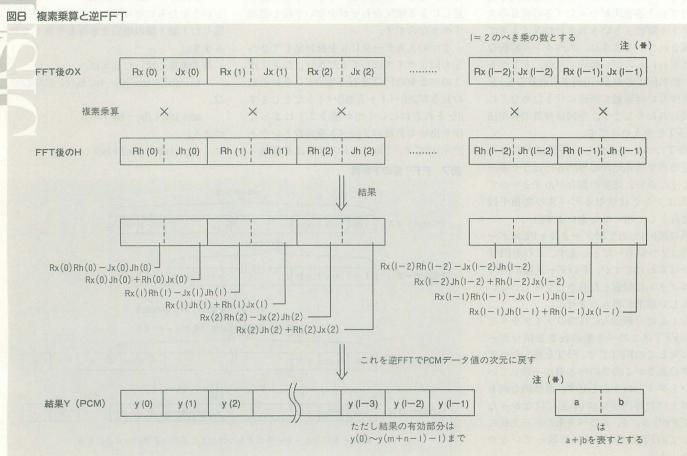
私は所有していないので詳しいことはわかりませんが「数値演算プロセッサ」があればこのあたりはある程度改善されるのかもしれません。

どのくらいの長さのデータだと，FFTのほうが早くなるかを条件判断して，整数演算を行うかFFTを使用するかを決定する……というのが理想でしょうか。

畳み込んで個性的な音作りを

音楽演奏にAD PCMを使用することが発明(?)されたときはたいへん度肝を抜かれたものです。そして，つい2年前からソフトウェア処理だけで多重再生が可能になりました。これからはPCM音をユーザー自身がデザインする時代が訪れようとしているのです。X68000の本体後面の「AUDIO-IN」端子は飾りではありません。ZVT.Xなどを駆使していろんな音をサンプリング，そして加工し，個性的な音をどんどん作っていきましょう。

[参考文献]
岩波情報科学辞典，岩波書店

図8 複素乗算と逆FFT

音楽データの標準形式

スタンダードMIDIファイルとはなにか

Nobata Hideaki 野畠 英明

コンピュータ用，楽器用に市販されている多くの音楽データがあります。その多くはスタンダードMIDIファイルというかたちで共通化されています。ここではスタンダードMIDIファイルの基本構造を紹介しましょう。

ここ数年の間にMIDIに関する環境は大きく変わってきました。以前は，DTMでMIDI音源といえば，ローランドのMT系のことを指していたものです。それがいまは同じくローランドではありますが，GS規格の音源（ここではあえてSC-55系とはいわないでおきます）が隆盛を誇っています。また，そのGS規格のベースになっているGM規格の音源も各社からたくさん発売されています。

このGM規格というのは，最低限の演奏情報の共通性を持たせようということで決められた規格で，主に音色配列などを規定しています。

ハードのほうでこのようにMIDI規格自体の枠より一歩踏み込んだ標準的な規格が決められたと同様に，ソフト（演奏ファイル）のほうでも標準的な規格が決められています。それが，スタンダードMIDIファイル（以下SMFと略記）です。日本でこそ，レコンポーザのRCPファイルが主流のようですが，世界的に見ればSMFが演奏ファイルの標準形式として広く利用されているようです。

SMFの用途

さて，SMFはどのように使われているのでしょうか。

まず挙げられるのは，各種演奏データ間のコンバートでの使用でしょう。さまざまな音楽ソフトが市販されていますが，SMFを読み書きできるものが多いので，いったんSMFとして書き出せば，どこへでも持っていけるわけです。

また，それと重複しますが，演奏データを流通させる場合を考えてみるとどうでしょう。この場合においては特に，パソコンやMIDI音源の機種や音楽ソフトに関係なく使用できるSMFの利用価値は大きいといえます。

たとえば，最近は楽器屋さんなどで，GM規格の音源用のSMFが3.5インチFDに入れられて販売されているのを多く見かけます。また，TAKERUや大手商用ネットなどでも同様にさまざまなSMFデータが販売されているようです。これら豊富なデータを活用しない手はありません。ただしその性格上，音源の能力をフルに駆使するようなものはあまりありませんから，聴くためだけに購入するのはどうかと思います。

別に販売するといったことでなくても，通信の世界などでは，特定ソフトの固有のデータ形式では多くの人に利用してもらうには不都合ということで，わざわざSMFに変換して配布している人もけっこういます。

あと，これは一般のユーザーには関係ないのですが，SMFはほとんどMIDIデータそのものといっていいような構造をしているので，演奏プログラムなどを作るのが比較的楽であるということもいえるでしょう。実際，SMFを演奏するプログラムはかなり多く見受けられるようです（なぜかX680x0ではあまり見かけませんが）。

SMFの構造

SMFはいくつかのブロックによって構成されています。ブロックは4バイトのタイプを示す文字と32ビットのブロックサイズ，それに続くサイズ分のデータからなっています。

現在のところブロックには，ヘッダブロックとトラックブロックの2種類があります。これは今後拡張される可能性がありますから，未知のブロックがあれば読み飛ばさなくてはいけません。ヘッダブロックは必ず先頭にあって，SMF全体に関する情報を格納しています。一方，トラックブロックは複数個あり，各演奏トラックのデータを格納しています。

トラックブロックではなかなか面白い数値の格納をする場合があります。これは可変長形式と呼ばれています（囲み記事参照）。

●ヘッダブロック

タイプを表す文字は「MThd」です。ここで，SMFのフォーマット，トラック数，分解能を指定します。

フォーマットには次の3つがあります。

0：1本のトラックのみ使用する
1：複数のトラックを使用する
2：互いに独立した複数トラックを使用する

フォーマット0は，ひとつのトラックにすべてのチャンネルのデータを格納してしまうもので，もっとも生のMIDIデータに近い形式といえます。そのため，MIDIボードから直接入出力するようなデータを扱うときに適しています。

また，フォーマット1は，MIDIチャンネルごとにトラックを割り当てるもので，Z-MUSICでのトラックの概念とよく似ています。たとえば，Z-MUSICのデータをSMFに変換するコンバータを作ろうといった場合にはこのフォーマットを使うのがいいと思います。

SMFのデータ集について

スタンダードMIDIファイルのデータ集は数多く市販されています。3.5インチ2DDのものはほとんどがMS-DOSフォーマット（9セクタ720Kバイト）です（ごく稀に3.5インチ2DDでもMacintoshフォーマットのものがあるので注意）。5インチ2HDというのはまずありません。

これらはX68030CompactまたはX68000CompactXVIを改造したもので，Human68k ver.3.0を使いFDDEVICE.Xを常駐させれば直接読み込むことができます。

GS/GM音源対応と書かれている場合は，ほぼSC-55で作成されたデータだと思っていいでしょう。GM音源用といっても機種によりかなり差があり，かなり違和感のある演奏になる場合もあります。注意してください。

フォーマット2はテンポなどの異なるトラックを共存できるもので，一般の演奏データではあまり使用しないものですし，対応しているものも少ない（と思う）ので詳しいことは省略します。

トラック数は文字どおりの意味で，たとえばフォーマット0ではもちろん1になりますし，それ以外なら1以上の数値になります。

最後に分解能ですが，もし正の数であるなら，4分音符の分解能に当たります。たとえば，Z-MUSICのデフォルトのモードは全音符が絶対音長で192ですから192/4＝48になります。負の数の場合はまた違った意味を持つのですが，一般の演奏データではまずないと思われますのでここでは省略します。

たとえば，フォーマット0で，分解能48の場合は次のようになります（16進数）。

4D 54 68 64	'MThd'
00 00 00 06	ブロック長
00 00	フォーマット0
00 01	トラック数1
00 30	分解能48

●トラックブロック

タイプを表す文字は「MTrk」です。このブロックには実際の演奏データが格納されます。この演奏データはデルタタイム＋イベントで1セットのデータが続いているものです。

デルタタイムとは，次のイベントまでの時間を表す数値で，可変長形式で格納されます。ヘッダブロックのところで述べた分解能はこのデルタタイムの分解能を示しているといえます。

イベントは，演奏情報を表す数値で大きく分けて，MIDIイベント，メタイベント，エクスクルーシブイベントの3つがあります。MIDIイベントというのは，ほぼMIDIデータと同じで，音符情報や，ボリューム，音色変更など演奏の中心となる情報を扱うイベントです。メタイベントは，MIDIデータ以外の情報を扱えるように定義されたイベントで，コメントや歌詞を格納したり，テンポを設定したりできます。

単に文章で説明されてもよくわからないと思いますから，まずMIDIイベントについて，簡単な例を示してみます。

たとえば，SC-55のチャンネル1で，ボリューム120，ベロシティ100でオクターブ4のドを2分音符の長さで発音する場合を考えてみましょう。まず，これをZ-MUSICのMMLで書くと，

(m1,100)(aMidi1,1)

(t1) @v120 @u100 o4 q8 c2

のようになります。今度は，これをSMFの格納方法に書き直してみましょう。分解能が48だとすれば，

00 B1 07 78	ボリューム
00 91 3C 64	ノートオン
60 91 3C 00	ノートオフ

というふうになります。最初の1バイトがデルタタイムで（可変長形式なので2～4バイトの場合もあります），それに続く3バイトがMIDIデータそのものであることがわかると思います。MIDIデータについては，MIDI楽器のマニュアルに詳しく載っているはずなのでそれを参照するとよいでしょう。注意すべき点は，デルタタイムはたとえ0でも省略できないということくらいでしょうか。

この例のように，基本的には先に述べたようにデルタタイム＋イベントというパターンで格納されるのですが，データサイズの削減のため，MIDIイベントに限って多少変形した形式が許されています。これは，ランニングステータスと呼ばれていて，ひとつ前と同じMIDIイベントの場合にステータスを省略してもよいというものです。

ランニングステータスを使って先の例を書き換えると，

00 B1 07 78

00 91 3C 64

60 3C 00

のようになります。どうです，1バイトの節約になっているでしょう。

ランニングステータスが使えるのは，MIDI規格では，ステータスが128以上，つまりバイト単位でのMSB（最上位ビット）が1なのに対して，それ以外のデータは127以下，つまりMSBが0だからです。なお，これは別にSMFに特有のものではなく，実際のMIDI楽器とのデータのやりとりにも使われます。

次に，メタイベントについてですが，このメタイベントというのはMIDIイベント

可変長形式とは？

可変長形式というのは，数値を効率よく格納しようという観点から考えられたもので，0～0FFFFFFF$_H$の数値を1～4バイトの数値として格納できます。

普通に数値を格納すると，バイト単位にすると1バイトで済みますが，0～255しか扱えません。また，じゃあロングワード単位にしようとすると，10とか20といった数値を格納するだけでも4バイト必要となり不経済です。

そこで，可変長形式では，0～127はそのまま1バイトで格納，それ以上の場合には7ビットごとに分けて，上位のほうから順にバイト単位でMSBを1にして格納し，最後のものだけ最後のバイトであることを表すためにMSBを0にして格納します。

これをそのまま考えるとどんな大きい数値でも格納できますが，4バイトまでということが決められています。これは，可変長形式で4バイト（通常の表し方で0FFFFFFF$_H$）あれば必要十分だからです。無限に大きい数値まで考慮しないといけないというのでは困りものですからね。

いくつか例を挙げておきます（左の数値を可変長形式にすると右のようになります。どちらも16進数です）。

00000000	→	00
00000010	→	10
0000007F	→	7F
00000080	→	81 00
00005000	→	81 A0 00
0FFFFFFF	→	FF FF FF 7F

また，可変長形式のデータを読み書きするアセンブラのマクロを載せておきますので参考にしてください（サブルーチンが好きな人はサブルーチンに直して利用するといいでしょう）。

リスト1

```
 1: *
 2: *
 3: *          可変長形式の数値を扱うマクロ
 4: *
 5: *
 6:
 7: *
 8: *          可変長形式のデータを読み出す
 9: *
10: *          ptr (アドレスレジスタ) のアドレスから
11: *          d0.l に読み出す。
12: *
13: GETVAL              macro    ptr
14:                     .local   GETVAL1
15:                     .local   GETVAL2
16:                     move.l   d1,-(sp)
17:                     moveq.l  #0,d0
18:                     moveq.l  #0,d1
19:                     move.b   (ptr)+,d1
20:                     bpl      GETVAL2
21: GETVAL1:
22:                     andi.b   #$7f,d1
23:                     or.b     d1,d0
24:                     lsl.l    #7,d0
25:                     move.b   (ptr)+,d1
26:                     bmi      GETVAL1
27: GETVAL2:
28:                     or.b     d1,d0
29:                     move.l   (sp)+,d1
30:                     endm
31:
32: *
33: *          可変長形式のデータを書き込む
34: *
35: *          ptr (アドレスレジスタ) のアドレスから
36: *          d0.l の内容を書き込む。
37: *
38: PUTVAL              macro    ptr
39:                     .local   PUTVAL1
40:                     .local   PUTVAL2
41:                     .local   PUTVAL3
42:                     .local   PUTVAL4
43:                     movem.l  d0-d1,-(sp)
44:                     moveq.l  #4-1,d1
45:                     rol.l    #4,d0
46: PUTVAL1:
47:                     rol.l    #7,d0
48:                     andi.b   #$7f,d0
49:                     bne      PUTVAL2
50:                     tst.l    d1
51:                     bgt      PUTVAL4
52: PUTVAL2:
53:                     bset.l   #31,d1
54:                     tst.w    d1
55:                     beq      PUTVAL3
56:                     ori.b    #$80,d0
57: PUTVAL3:
58:                     move.b   d0,(ptr)+
59: PUTVAL4:
60:                     dbra     d1,PUTVAL1
61:                     movem.l  (sp)+,d0-d1
62:                     endm
```

とはちょっと違った形式をしています。例として，トラックの終わりを示すメタイベントを以下に示します。

FF 2F 00

最初のFFはメタイベントを示すもので，すべてのメタイベントはこれで始まります。次の2Fはメタイベントのタイプを示すものです。最後の00はこの後ろに続くデータの長さで，可変長形式で表します。この場合は後ろに続くデータはありませんから0なわけですが，ほかの多くのタイプでは0ではありません。ここで注意したいのは，このデータの長さを無視してはいけないということです。通常は，メタイベントによっては無視しても支障はありませんが，将来変更があるかもしれませんし，そうでなくともソフトのバグなどでちょっと大きめの長さになっていないとも限りません。想定した長さより大きい場合は無視するのが正しい処理といえます。この例ではデルタタイムを省きましたが，本当は必要ですので勘違いしないように。ほかのメタイベントについては別表を参照してください。

さて，残りのエクスクルーシブイベントですが，これは2つあります。なんで2つあるかというと，MIDI楽器によっては，ひとつのエクスクルーシブを複数のパケットに分けて，適当なウエイトをかけて送ってやらないといけないから，ということらしいです。困ったもんですね。たぶん，これを読んでいる方は，そんな面倒なことをしなくてもいいMIDI楽器を使用しているだろうと勝手に解釈して，ここでは主に一方だけ解説します（んなわけないだろーとか思った人，ごめんなさい)。このイベントはMIDIイベントと同じようにほとんどMIDIデータと同じです。ただ，その長さに関する情報が加わっているにすぎません。

たとえば，Z-MUSICのSC-55用データでおまじないのように最初に実行されるオールパラメータリセットを例にとると，

F0 0A

41 10 42 12 40 00 7F 00 41 F7

のようになります。最初のF0がエクスクルーシブイベントであることを示し，その次の0Aがそれに続くエクスクルーシブデータの長さを表します。長さはもちろん可変長形式です。また，くどいようですが，実際にはこの前にデルタタイムがつきます(耳にタコができたかもしれませんけど)。

まったく触れないのもなんなので，もうひとつの形式についても少し説明します。イベントコードがF7であるほかは通常のエクスクルーシブイベントと変わりありません。ひとつのエクスクルーシブデータを必要なパケット数に分割し，最初のひとつを先の通常の場合のイベントで，残りをいま述べたイベントで送ればいいわけです。ウエイトはデルタタイムで調節することになります。

最後に

これまでの説明で，SMFについてある程度は理解していただけたのではないかと思います。

MIDI規格自体もずいぶん古い規格ですから，それを元にしたSMFもいまの時代にはいささか不便な点もなくはありません。

けれど，最初のほうで述べたように世界標準であるこの形式の利用価値は十分あるといえるでしょう。

これまで，SMFを知らなかったという人や名前は知っていたけど利用していなかった人もこれを機会に利用してみてはどうでしょうか。

メタイベント一覧

ここでは，簡単な説明しかしませんので，詳しいことはそれなりの文献をあたってください。

●シーケンスナンバー

FF 00 02 num

このイベントは，複数のシーケンスを用いる場合に，各々を識別するためにあるようです。それから，トラックの冒頭，任意の0でないデルタタイムの前で，任意のMIDIイベントの前に置かれなければなりません。

●テキストイベント

FF 01 length text

トラック中で任意のコメントなどを格納するのに用います。曲のタイトルやデータの作者名などを格納するのにもこのイベントを用いるようです。

lengthはtextの長さで，可変長形式で格納されます。これは以降のメタイベントでも同様ですので，これ以降省略します。

●著作権表示

FF 02 length text

著作権を，ASCIIテキストとして持ちます。この表示には(C)の文字と，著作物発行年と，著作権所有者名とが含まれなければなりません。また，最初のトラックブロックの最初のイベントとして，デルタタイム0で置く必要があります。

●シーケンス名（トラック名)

FF 03 length text

最初のトラックに置いてシーケンス名やトラック名を格納します。

●楽器名

FF 04 length text

そのトラックで用いる楽器の種類などを格納します。

●歌詞

FF 05 length text

歌詞を格納します。

●マーカー

FF 06 length text

最初のトラックに置かれ，その時点でのリハーサル記号やセクション名のようなものを格納します。

●キューポイント

FF 07 length text

スコアのその位置においてフィルムやステージ上で起こっていることを記述します（たとえば，この部分を演奏しているときにステージ上ではなにをやっているのかなどといったことをコメントとして残したい場合に用いる)。

●MIDIチャンネルプリフィクス

FF 20 01 ch

MIDIチャンネルを指定します。これによって，チャンネル指定のできないイベントでも，チャンネルを指定することができます。

●トラック終端

FF 2F 00

このイベントでトラックの終わりを示します。したがって，このイベントは省略することができません。

●テンポ設定

FF 51 03 tempo

最初のトラックに置かれ，テンポの設定を（μsec/四分音符）という単位でします。これを音楽的テンポに直すには（1000×1000/tempo×60）という計算をすればよいわけです。

●SMPTEオフセット

FF 54 05 h m s f f

トラックブロックがスタートすることになっているSMPTEタイムを示します。このイベントは，トラックの冒頭，任意の0でないデルタタイムの前で，任意のMIDIイベントの前に置かれなければなりません。

●拍子記号

FF 58 04 n d c b

d分のn拍子という拍子を格納します。また，cは1メトロノームあたりのMIDIクロック数を，bはMIDIの4分音符中に記譜上の32分音符がいくつ入るかを示します。

●調号

FF 59 02 sf mi

sfでシャープやフラットの数（プラスならシャープ）およびmiで長調（0）か短調（1）かを格納します。

●シーケンサ特定メタイベント

FF 7F len data

特定のシーケンサのための特別なイベントとして使用されます。なお，1バイト目はメーカーIDを格納します。

（備考）

タイプで01～0Fは，未定義のものも含めて基本的にテキストデータを格納するイベントのために使用されます。

［特集］
Z-MUSICシステム
ver. 2.0

演奏データ作成ガイドライン

ポータビリティの高いデータとは

Tama Tamaki　たま たまき

ポータビリティ。すなわち，「ほかの環境でもなるべく問題なく演奏できること」に重点を置いたデータの作り方を解説します。せっかく作成した音楽データですから，できるだけ広範囲に活用したいものですね。

昨年は各社からGM対応音源が発売されて選択肢が増えたのはよいことなのですが，すべてのGM対応音源で正しく演奏できる演奏データを作成するのは難しいものです。たとえば，エフェクタについてもローランドのGS音源であればコーラスとリバーブが使用できますが，GM音源のなかにはリバーブすらついていない音源だって存在します。

GSフォーマットというのは確かにGMを包括した規格ではありますが，極端な例を挙げるとSC-55のキャピタルだけを使用して作成した演奏データでもエフェクタをバリバリ使用していたとしたら，エフェクタのないQY-20のGMエミュレーションモードではどうなるでしょうか？　エフェクタの効果が得られないのは当然ですが，最悪の場合，曲全体のボリュームバランスが崩れてしまい，とても聞けたものではないといったことさえ起こります。しかし，この演奏データはGMの規格の範囲内のことしかやっていないのです。

演奏データにポータビリティ（可搬性：要するに別の環境でも動作すること）を持たせようとすると，想定される最少構成の音源を前提に作成しなければならないのです。そんなことは素人にはとてもできません。

X68000ではむしろ音源を特定して，音源の性能限界に迫るようなミュージックデータが中心になっていますが，世間ではより広い範囲で使える音楽データの普及に力を入れている向きもあります。もともとSMFやGS，GMという規格はそのために作られたようなものです。

ということで，今回はポータビリティの高い演奏データ作成法について，筆者が普段から心掛けている点を解説したいと思います。最初にお断りしておきますが，本文は「GSを前提」に記述しています。ほかの音源に関しては，私が気がついた分については注記しましたが，使用できないコントロールもあるかもしれませんので，詳細はお手持ちの音源のMIDIインプリメントチャートで確認してください。

データ作成上の注意点

1)　エクスクルーシブと演奏データは分離する

X68000ユーザーの皆さんはどちらかというとMMLで演奏データを作成する人のほうが多いと思います。MMLで作成されている演奏データはエクスクルーシブと演奏データは分離することができるようですが，世の主流のスタンダードMIDIフォーマットなどでは，当然のごとく演奏データにエクスクルーシブを埋め込んでいます。データの扱いという観点から見れば便利ですが，これにより最大の機種依存が発生するのですからポータビリティを求めるのであれば分離すべきです。

2)　曲の途中でプログラムチェンジはなるべく使用しない

これは「えっ」と思われる方がおられると思いますが，曲中のプログラムチェンジの置き換えというのは結構労力がいるものです。GSやGMであれば同じプログラムチェンジで同系列の音が割り振られていますからよいかもしれませんが，たとえば，GS用の演奏データをMT-32などに手直しする際は当然，プログラムチェンジを置き換える作業が必要となるでしょう。

また，曲のアレンジを変更する際に「ちょっと違う音にしてみよう」と思っても，曲中にプログラムチェンジがあると書き換えるのは面倒です。ましてや，他人が作成した演奏データだったらプログラムチェンジを探すのもひと苦労です。

これらの理由から，使用するシーケンスソフトのトラックの許す限り，1トラック1音色の原則を守って，曲中でプログラムチェンジは使用しないほうがよいでしょう。

やむをえず曲中でプログラムチェンジを使用する場合はプログラムチェンジの後ろに4分休符程度の間隔を空けてから発音するように心掛けてください。音源はプログラムチェンジを受信したパートは一定期間発音できない状態になるのです。

この一定期間というのが音源によってまちまちですが，4分休符程度の間を取っておけばかなりレスポンスが遅い音源でテンポの速い曲を演奏した場合でも大丈夫でしょう。逆にテンポ160で4分休符も間を取ったにもかかわらず音切れするような音源はあきれてものがいえません。

3)　ボリュームとエクスプレッションを使い分ける

コントロール7番のボリュームとコントロール11番のエクスプレッションはともに音量を調整するものですが，これを分けることによって，音量の調整にポータビリティを持たせることができます。

ボリュームとエクスプレッションの関係ですが，ボリュームがメインコントロール，エクスプレッションがサブコントロールだと思ってください。

つまりボリュームを100に設定した場合，エクスプレッションで0を送信すれば音は出力されず，127を送信すれば本来のボリューム100の音量で出力されるというわけです。

多くの人はボリュームだけで音量をコントロールしています（つまり，エクスプレッションは127のまま）が，曲の最初で最大音量を決定するためだけにボリュームを送信し，音量の増減をエクスプレッションでコントロールするようにしておけば，あとで全体のバランス取りの際に個々のパートの音量の調整は曲の先頭に書いたボリュームの値だけを変更すればよいのです。

また，エクスプレッションでコントロールしておけば，演奏中に音源側のボリュー

ムコントロールを変更しても，変更後のボリューム値からの比率で変化するわけですから，データ作成者の意図したとおりのボリューム調整がなされます。これは結構便利ですよ。

注）MT-32やGS音源ではエクスプレッションは有効ですが，エクスプレッションがサポートされていない音源もあります。U-220やSY-77などはエクスプレッションをサポートしておりませんが，コントロールを定義することができますのでエクスプレッションを受信したらボリュームを変更するように設定すればよいでしょう（ただし，サブボリュームとしては機能しない）。

4）先頭にセットアップ用小節を用意する

これは1），2），3)で述べたことに関係することですが，曲の先頭にセットアップ用の小節を用意しておいて，プログラムチェンジや短いエクスクルーシブなどはここにすべて記述するようにします。これによって，演奏データにエクスクルーシブを埋め込んでも取り除くのは容易になりますし，音色を変更するのも簡単です。

また，エクスクルーシブと演奏データを分離した場合，エクスクルーシブを送信したあと少し間をおいておかないと曲の頭が欠けてしまうこともあります。こういう場合にも1小節余裕があれば十分でしょう。

5）タイムベースは1/96が基本

スタンダードMIDIフォーマットの場合，多くのシーケンスソフトやシーケンサは1/96のタイムベースはサポートしているハズです。パソコン通信で演奏データを配布したいと考えているのであれば，必ずは1/96にすべきでしょう。

楽器業界ではタイムベースは1/96を標準としているようです。

ちなみにローランドのシーケンサでスタンダードMIDIフォーマットをサポートしている機種は1/96であれば演奏できることを保証しているようですし，SY-99の内蔵シーケンサのタイムベースも1/96です。

Z-MUSICではタイムベース48が基本となっていますから，特に気を遣うことはないのですが（低い分には問題はない），Mu-1などでリアルタイム入力したデータなどはしっかりクォンタイズしておいたほうがいいでしょう。

6）特殊効果は専用トラックで

これは2)や3)に関連したことですが，ポルタメントなどRPNやNRPNのメッセージは結構重たいので，曲中に送信すると音切れの原因になります。また，エクスクルーシブを送信して設定する特殊効果もあるでしょう。

専用トラックを用意しておけば音源への負担はかなり軽くなりますし，ポータビリティも高まるでしょう。

7）リズムパートを優先

リズムパートはできるだけ若いトラック番号に作成し，優先して出力するようにするとよいでしょう。なぜなら，GSフォーマットではデフォルトでパート10がリズムパートに割り振られていますが，パート10がいちばん優先順位が高いのです。

また，リズムパートの音切れは気になるものです。

パートの割り振り方としては，GSのデフォルトのパーシャルリザーブの設定からパート1には比較的発音数が多いピアノやギターなどのメインとなるコードを演奏するもの，あとはパート2～6を優先的に使用するのが望ましいでしょう。

8）音の強弱はベロシティで

よくボリュームとベロシティの違いがわからず，音の強弱をすべてボリュームで指定している演奏データを見かけます。これはすごくナンセンスです。

ボリュームというのは単に音量を設定するだけで音質は変化しませんが，ベロシティは音量とともに音質も変化します。音の強弱には音質という要素も含まれます。

たとえば，メゾピアノとかフォルテなどの強弱記号はベロシティで表現します。

逆にエレキギターのバイオリン奏法のようなクレシェンド，デクレシェンドを表現する場合はボリュームでコントロールしたほうがよいでしょう。バイオリン奏法の場合，ひとつの音符で音量をコントロールしなければならず，ベロシティでは表現できないからです。同じクレシェンド，デクレシェンドでも比較的短い音符が連符している場合はベロシティで表現します。このへんを使い分けてください。

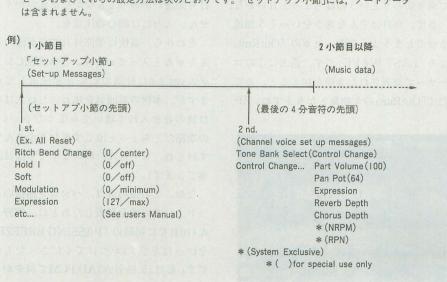
おわりに

以上，演奏データ作成について常識的なことを中心に書きました。参考になるデータとしては，ローランドから発売されているSMFシリーズが上記のガイドラインに沿って作成されています。SMFのデータはかなりポータビリティが高く，シーケンサソフトで中を眺めるのも結構勉強になりますので，ひとつくらいは教材だと思って買ってみるのもよいでしょう。X68000ではMu-1Superでそのまま演奏できます。Z-MUSICでもMZP.Xを使えば演奏できます。ただし3.5インチ2DDドライブが必要になりますが……。

また，打ち込み技術を向上させるには，他人が打ち込んだデータの中のテクニックを盗むことがいちばんでしょう。プログラミングの勉強とさほど変わりはありません。そういった意味でもNIFTY-ServeのFMIDIは重宝しますので，NIFTYのIDをお持ちの方はFMIDIに入会してみるとよいでしょう（ちなみに筆者はNIFTYやFMIDIとは関係ありません）。

それでは，楽しいDTMライフを。

表1　セットアップ小節

曲の最初に，システム初期化のためのエクスクルーシブメッセージや，ほかの「セットアップ」メッセージのみを含む「セットアップ小節」を設けてください。この中に含まれるメッセージおよびそれらの設定方法は次のとおりです。「セットアップ小節」には，ノートデータは含まれません。

Oh!X LIVE in '94

X68000・Z-MUSIC用	©SEGA「OutRun」より **LAST WAVE**	Shindo Noriyuki 進藤 慶到
X68000・Z-MUSIC用(SC-55対応)	**スターウォーズ**	Sasaki Tsugutomo 佐々木 嗣朋
X68000・Z-MUSIC用(SC-55対応)	**明日への扉**	Tanabe Masanori 田辺 正則
X68000・Z-MUSIC用	**夢路より**	Kato Takashi 加藤 隆
X1・MusicBASIC用	©NAMCO ALL RIGHTS RESERVED **NEW RALLY-X**	Yamada Miho 山田 美保

今年は戌年。ってのは特に関係ないけど盛りだくさんの5曲で始まるLIVE in '94。「OutRun」シリーズの完結や久々にX1登場など内容も盛りだくさん。Z-MUSICの新バージョンも発売開始。今年もたくさんの投稿やご意見お待ちしています。

これでめでたく全曲集

皆さんあけましておめでとうございます。最近，体が2つ欲しい進藤です。

それにしても寒いですね。これを書いているいまはまだ11月なのですが，こんな調子ではこれから先が思いやられます。1993年の夏はエアコンなんか要らねーよってくらい涼しかった東京ですが，やはりいつもの冬がやって来るようで……。コタツだけで乗り切ろうというのはちょっと甘いかな。

さて，今月はそんな寒さをいっそう加速させてしまうような曲，セガの「OutRun」より「LAST WAVE」です。過去にこのコーナーで発表された3曲を加えると，めでたく「OutRun」の全曲集となりますね。ZP.Xのジュークボックスなどでもお楽しみいただけます。

今回のリストは，私の作ったものとしてはいつになく短いです(曲が短いんだから当たり前……)。ただ，メロディパートだけは姑息なステレオ技を使用したことと，音の重なりを出すための苦肉の2トラック処理のおかげで(Z-MUSICの弱点かも)，一部非常に見づらくなってしまいました。トータルステップカウントを出せば長さが合っているかの確認はできますが，それ以外の打ち間違いがあってもなかなか見つかりません。入力には細心の注意を。

それから，豪快に季節外れですが波の音もちゃんと入っています。Z-MUSICシステムver.2.0に付属の効果音を使用していますが，本物の雰囲気を味わいたい人はぜひ波の音を入れて聴いてみてください(この季節だとちょっと涼しすぎたりするんですけどね。もちろん，波がなくても曲は演奏できます)。

なお，この波の音は，パソコン通信のネット上にこの曲を発表したあとに，舘野さん(10月号に掲載の「PASSING BREEZE」を作った方です)がつけてくださったものです。私は15.6kHzのAD PCMで爽やかな波の音は出せない！ と思い込んでいたのですが，意外とそうでもないんですね。「AD PCMあなどり難し」といわんばかりの，なかなかいい音が聴けます。波のMMLを書いてくれた舘野さんには，この場を借りてお礼申し上げます。 (進藤慶到)

OutRun

フォースがともにあらんことを

さて，LIVE inも'93から'94になり，気分一新でがんばりましょう。ということで，年明けの2曲目です。「may the force be with you」といえば，知らない人でも知っているといわれる(?)，映画「スターウォーズ」の名セリフですね。X68000シリーズにも驚異の3D処理をした同名のゲームがあり，お馴染みですよね。今回紹介する曲はサントラ版より「エンドタイトル」です。

スターウォーズ

前出のゲームではタイム表示〜ネーム入れのあたりでかかる曲ですね。試しに聴き比べてみましたが，相手が内蔵音源とはいえ，まるで比べものにならないほどの仕上がりです。そうそう，演奏にはSC-55もしくは同等品が必要です。

この作品では実際には第２楽章までしかないそうですが，十分な出来栄えを誇っています。リストもそこそこ短くまとまっているので，よい選択かもしれません。また，リストの最後にある［loop］を有効にすると無限ループになってくれます。

かなりゴージャスなオーケストレーションですので，「SC買ってよかったぁ〜」なんて思うこと請け合い。ティンパニの音などはあまり使うものではありませんので，こういった機会に堪能するとよいですね。

作者の佐々木君は1993年９月号以来ですね。これからも常連目指してがんばってくださいね。

どこでもドアぢゃないよね?

さて３曲目もちゃおちゃおっと紹介しちゃおうっと。アルバム「HUMAN」から「明日への扉」をお届けしましょう。「HUMAN」といっても残念ながら「〜68k」ではありません。念のため。そう，お馴染みT-SQUAREのアルバムなのです。この曲はフジテレビ系のF1グランプリでも挿入歌として使われていました。

この曲もデキはいいですね。特にオープニングから絶妙な雰囲気を見事に再現しています。さすがにCDクオリティとまではいいませんが，聴かなきゃ損ソン，ってカンジです。もうすこしメリハリをつけることができれば，かなり「きてる」作品に仕上がったと思えます。過去にLIVE inを飾ったフュージョン系の曲のレベルの高さを知っている人なら，いわずとももう入力を始めていますよね？

1富士2鷹3ナスビ

……といえば初夢ですね。なんて１月号にふさわしいネタなのでせう。そこで，フォスター作曲の「Beautiful Dreamer」をお届けしましょう。邦題では「夢路より」あるいは「夢みる人」ですよね。フォスター

HUMAN

といっても，ジョディでないことくらいはみなさんご存じでしょう。「Yes CIVIC, Yes FERIO」とは関係ないということです。この曲，おそらく多くの人は小・中学校あたりの音楽の時間で歌ったことがあるでしょう。「たまごからプロテア〜」という歌詞はあまりにも有名ですよね(?)。

演奏にはZ-MUSICシステムが必要です。音色はピアノのみです。

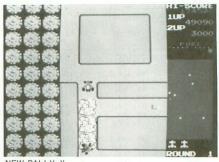
ピアノ曲では，ごまかしがきかないだけに技術力が求められます。この作品ではテンポを変えることで表現しているようですね。同じ加藤君の作品ですが，ボリュームも細かく変化させていた1993年11月号の「渚のアデリーヌ」のほうがより突っ込んでいたともいえるでしょうか。ただ，必ずしもテクニックだけに走った作品が素晴らしくなるとは限らないので，耳に心地よい音を探求してみてください。加藤君は最近，ピアノの表現に凝っているようですね。いずれ超絶技巧とも呼べるような作品を送ってくれることでしょう。

それにしても，この曲を作るきっかけは「金髪のジェニーが終わってしまったから」なんだそうです。そこはかとなく世紀末ですね（笑）。

史上初！　初春の珍事!?

さて，トリをつとめるこの作品はなんといっても久しぶりのX1のMusicBASIC用の作品です。よぉ〜く指をかっぽじって(?)入力してくださいね。曲は「NEW RALLY-X」のオリジナルアレンジバージョンです。「NEW RALLY-X」は10年くらい前のナムコのアーケードゲームです。知らない人はいませんよね？

FM音源＋PSGの構成ですので，ミキシングが必要な人はミキシングしてください。

夢みる人

NEW RALLY-X

えっ？「ところで，なにが史上初なの？」ですって。よくぞ聞いてくれましたっ！1987年に産声をあげたこのLIVE inのページ。当時はまだ「Oh！MZ」という雑誌でした。あれから苦節６年，紹介した曲も200曲を軽く超えています。……で，史上初とは……なんと初の女性採用なのです！　100人以上の採用者がいたにもかかわらず，なぜか女性はひとりもいなかったのです。すごいことですねー。別に女人禁制ではなかったのです。ここだけの話ですが，作者の美保ちゃんは018h歳です。ファンクラブとかできたりして……。

超長い導入でしたが，作品はかわいらしくまとまっています。とてもMML歴数週間の作品ではありません。ファンクラブ会員を名乗ろうと思っている君！　まず入力して，ちゃんと感想文をつけることっ！彼女はX68000進出計画もあるということですので，今後がとっても楽しみですね。

こいつぁ初春から縁起がいいやっ！　ってそれはともかく，せっかくの正月休みを家で過ごすなら，MMLとつき合うというのも楽しいかもしれません。作品ができあがったら，投稿してくださいね。

さて，５曲掲載という豪華版で幕を開けたLIVE in '94。今年もハイクオリティな作品をビシバシ掲載していきます。よろしくっ！

(SIVA)

リスト1 LAST WAVE

```
  1: .comment -OUT RUN- LAST WAVE (C)SEGA  by ENG 93/05/05 +
  2:
  3: / for ZMUSIC.X
  4: / Special thanks to TTN
  5:
  6: /------------------------------------------------------------
  7: / TRACK SETUP
  8:
  9: (i)
 10:
 11: / OPM & ADPCM
 12:
 13: (m1,1500)(aFm1,1)
 14: (m2,1500)(aFm2,2)
 15: (m3,1500)(aFm3,3)
 16: (m4,1500)(aFm4,4)
 17: (m5,1500)(aFm5,5)
 18:
 19: (m6,1000)(aAdpcm,6)
 20:
 21: /------------------------------------------------------------
 22: / ADPCM DATA SET
 23:
 24: .adpcm_block_data = LA_WAVE
 25:
 26: /------------------------------------------------------------
 27: / OPM DATA SET
 28:
 29: /        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  E.P.
 30: (@1,     31, 14,  0,  0, 15, 60,  0, 15,  5,  0,  0
 31:          31, 14,  2,  0,  1, 44,  0,  1,  1,  0,  1
 32:          31, 14,  4,  0,  1, 49,  0,  1,  0,  0,  1
 33:          31, 14,  7,  6,  1,  0,  0,  1,  2,  0,  1
 34: /        AL  FB  OM PAN  WF  SY  SP PMD AMD PMS AMS
 35:           2,  7, 15,  3,  2,  1,191,  0,  3,  0,  1)
 36:
 37: /        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  E.P.L
 38: (@2,     31, 14,  0,  5, 15, 60,  0, 15,  5,  0,  0
 39:          31, 14,  2,  5,  1, 44,  0,  1,  1,  0,  1
 40:          31, 14,  4,  5,  1, 49,  0,  1,  0,  0,  1
 41:          31, 14,  7,  4,  1,  0,  0,  1,  2,  0,  1
 42: /        AL  FB  OM PAN  WF  SY  SP PMD AMD PMS AMS
 43:           2,  7, 15,  3,  2,  1,191,  0,  3,  0,  1)
 44:
 45: /        AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  E.P.S
 46: (@3,     31, 14,  0, 15, 15, 60,  0, 15,  5,  0,  0
 47:          31, 14,  2, 15,  1, 44,  0,  1,  1,  0,  1
 48:          31, 14,  4, 15,  1, 49,  0,  1,  0,  0,  1
 49:          31, 14,  7, 15,  1,  0,  0,  1,  2,  0,  1
 50: /        AL  FB  OM PAN  WF  SY  SP PMD AMD PMS AMS
 51:           2,  7, 15,  3,  2,  1,191,  0,  3,  0,  1)
 52:
 53: /------------------------------------------------------------
 54: / MML DATA SET
 55:
 56: (o63)
 57:
 58: (t1)     @v0r*434L8@1o2@v121@k00q8t63
 59:          f2.fce2.e4r*348b.d2g4r<d>c2a4r<e>d2g4r<d>c2ccc16c.
 60:          L16|:ff_23f_10f_ff~43f8^2
 61:          |ee_23e_10e_ee~43e8^2:|e8e_23e_10e_e~43e8^2
 62:          L8d2g2c2a2d2g2cc_23c16_10c16~33d2>a<L16
 63:          |:3c8c_23c_10c_c~43|d8^2:|_1d*336c*384r1
 64:
 65: (t2)     @v0r*434L8@1o3@v120@k00q8
 66:          f2.fce2.e4~4f2.fce4.e4._17e~13
 67:          er4d4g4g4c2a4a4d2g4g4c2ccc16c.
 68:          L16|:ff_23f_10f_ff~43f8^2
 69:          |ee_23e_10e_ee~43e8^2:|e8e_23e_10e_e~43e8^2
 70:          L8d2g4gdc2a4aed2g4gdcc_23c16_10c16~33d2>a<L16
 71:          |:3c8c_23c_10c_c~43|d8^2:|_1d*336c*384r1
 72:
 73: /--------------------------------
 74:
 75: (t3)     @v0r*428L16@1o5@v124@k00@q1
 76:          p1d*1&p3d*2r*3
 77:          rp1e*1&p3e*23p1e*1&p3e*77r*3p1d+*1&p3d+*2
 78:          p1e*1&p3e*47p1c*1&p3c*21p1f+*1&p3f+*1
 79:          p1g*1&p3g*71r*9p1d+*1&p3d+*2p1e*1&p3e*59
 80:          r8p1c*1&p3c*17p1d*1&p3d*2r*3
 81:          rp1e*1&p3e*23p1e*1&p3e*77r*3p1d+*1&p3d+*2
 82:          p1e*1&p3e*47p1c*1&p3c*17>r*3p1a+*1&p3a+*2
 83:          p1b*1&p3b*47r8p1g*1&p3g*23@v0r2@v124
 84:          p1b*1&p3b*47<r12p1c*1&p3c*31p1d*1&p3d*35
 85:          >rp1a*1&p3a*31<p1d*1&p3d*15
 86:          r8.p1e*1&p3e*83>r<p1d-*1&p3d-*23
 87:          p1b-*1&p3b-*29p1g*1&p3g*5
 88:          r*72p1e*1&p3e*23r*57p1f+*1&p3f+*8r*18p1f+*1&p3f+*23
 89:          r4|:p1a*1&p3a*23r:|<p1c*1&p3c*35rp1e*1&p3e*8r*3
 90:          rp1e*1&p3e*23p1e*1&p3e*83p1d*1&p3d*22p1d*1&p3d*47
 91:          p1g*1&p3g*71rp1e*1&p3e*59r>r<p1c*1&p3c*23
 92:          p1e*1&p3e*23p1d*1&p3d*30r*65p1e*1&p3e*47p1c*1&p3
 93:          c*23r*64q4p1b*1&p3b*31p1b*1&p3b*35@q1r*60
 94:          @2p1e*1&p3e*23_12r~p1d*1&p3d*23_r8~p1c*1&p3c*23_r8>
 95:          ~p1b*1&p3b*23_r*20~@1q8p1g+*1&p3g+*15@q1
 96:          <p1a*1&p3a*35p1g*1&p3g*75q7p1b-*1&p3b-*31
 97:          p1g*1&p3g*31@q18p1e*1&p3e*31@q1
 98:          r*44p1d*1&p3d*35p1d*1&p3d*83p1c*1&p3c*83r*108
 99:          >p1a*1&p3a*11
100:          |:p1g*1&p3g*95r*84p1a*1&p3a*11:|p1g*1&p3g*47r*936
101:
102: (t4)     @v0r*428L16@1o5@v124@k00@q1
103:          r*3p1d+*1&p3d+*2
104:          p1e*1&p3e*23p1d*1&p3d*89p1d*1&p3d*2r*3r8
105:          p1d*1&p3d*42p1e*1&p3e*2r*2
106:          r4.p1d*1&p3d*8r*3r*60r>p1a*1&p3a*23
107:          <p1d*1&p3d*8p1d+*1&p3d+*2
108:          p1e*1&p3e*23p1d*1&p3d*89p1r*30p1d*1&p3d*41
109:          >p1a*1&p3a*2r*3
110:          r4p1b*1&p3b*35p1e*1&p3e*59&_13e4~
111:          r8.<p1d*1&p3d*43p1e*1&p3e*15p1e*1&p3e*11r>
112:          p1g*1&p3g*39<p1c*1&p3c*31
113:          p1d*1&p3d*47r*60>p1b-*1&p3b-*35
114:          <p1d*1&p3d*23rp1a*1&p3a*11
115:          p1f*1&p3f*83p1f*1&p3f*68r*3p1g*1&p3g*35p1g*1&p3g*71
116:          rp1g*1&p3g*23p1b-*1&p3b-*35r<p1d*1&p3d*32d*3
117:          p1e*1&p3e*23p1d*1&p3d*32r*60p1d*1&p3d*2p1e*1&p3e*47
118:          p1c*1&p3c*23r4.p1d*1&p3d*23r4r>p1a*1&p3a*23<
119:          p1d*1&p3d*11
120:          rp1e*1&p3e*23p1e*1&p3e*83r8p1d*1&p3d*47
121:          p1g*1&p3g*79q4p1b*1&p3b*27@q1p1g*1&p3g*83
122:          r8@2p1d*1&p3d*23_28p2d8~p1c*1&p3c*23_p2c8~>
123:          p1b*1&p3b*23_p2b8~@1@q5p1g*1&p3g*15p1a*1&p3a*7@q1
124:          <r8p1g*1&p3g*11r^4q7p1e*1&p3e*31p1a*1&p3a*15r*16
125:          p1f*1&p3f*31@q1
126:          p1d*1&p3d*59r8p1e*1&p3e*23r*60p1c*1&p3c*11r*81
127:          p1c+*1&p3c+*2p1d*1&p3d*113>
128:          |:p1b*1&p3b*7r*70p1d*1&p3d*113:|p1b*1&p3b*5
129:          r4r8<p1@q420d*1&p3d*719r1
130:
131: /--------------------------------
132:
133: (t5)     @v0r*427L16@2o4@v120@k00q8
134:          @d1f*5a*4<c*4e*114>@d0'fa'24'fa<c'24'a<ce'19
135:          'egb<d'77,4'eb<e'24,0'egb'36'eg''eg''e<c''e<d'15
136:          'fa<ce'129,3'fa<e'24,0r4'egb'48
137:          'eg<c'24&<c*0>&@1'egb<c'24&'egb'48_17g4~
138:          @2'dfa'48r12'fa<c'16'fa<e'16
139:          'egb<d'48'egb'48'egb<d'60@d1eg<d8.>
140:          @d0'eg'24'eg<c+'24'g<e'24
141:          'fa<ce'72'fa<d'72@d1b8<d8>@d0
142:          'egb<d'36'egb<e'36'eg'24'egb-'24
143:          'egb-<c'24'eg<ce''eg<ce'36
144:          @3'fa<ce''fa<ce'_23'fa<ce'_10'fa<ce'_
145:          |:'fa<ce':|~43'fa<ce'120
146:          'egb<d''egb<d'_23'egb<d'_10'egb<d'_
147:          |:'egb<d':|~43@2'egb<d'72@d1gb<d8>@d0
148:          @3'fa<ce''fa<ce'_23'fa<ce'_10'fa<ce'_
149:          |:'fa<ce':|~43'fa<ce'120
150:          'egb<d'24'egb<d'_23'egb<d'_10'egb<d'_
151:          'egb<d'~43@2'egb<d'60@d1egb<d8>@d0
152:          'dfa<c'48_23'dfa<c'~'dfa<c'36'fgb<d'48'fgb'48
153:          'egb<d'72<d8>'c+ea'36@d1c+32e32g<c+e8@d0>
154:          'fa<ce'60f8.'fgb<d'36f'fgb<d'48@3
155:          'egb<d'24'egb<d'24_23'egb<d'_10'egb<d'~33'dfa<c'120
156:          |:3'egb<d'24'egb<d'_23'egb<d'_10'egb<d'_'egb<d'~43
157:          |'dfa<c'120:|@2'dfa<c'72
158:          @d1dfa*14<c*15e*16g*18a*19<c*18&c*140>>
159:          _1c*5e*4g*4b*371@d0r1
160:
161: /--------------------------------
162: / ADPCM WAVE   Featuring TTN
163:
164: (t6)     @2@f4o6L*117¥-45r*38_18~
165:          v2cv5cv7cv9c_d_cL*116~~cdcd_d_d~~ccd_d_d_d~~~
166:          L*117c_d~d_c_c_c~~~c_c~L*116dc_c_c~~cdc_c_d_dv10cv9
167:          L*115cv10dv8d_c_c@3v10c+_d_c
168:
169: (p)
```

リスト2 LAST WAVE用音色コンフィグファイル

```
/ -OUT RUN- LAST WAVE (C)SEGA  by ENG (+PCM8)

.adpcm_bank 2

.o6c = wave3.pcm,v26,c0,20000
.o6c+= .o6c,p-2
.o6d = .o6c,p2,f15000,80
```

リスト3 LAST WAVE用カウンタ表示

```
1:0000130A 00000000  2:0000130A 00000000  3:00001309 00000000  4:0000130A 00000000
5:0000130A 00000000  6:00001490 00000000  7:00000000 00000000
```

リスト4 スターウォーズ


```
1: /
2: /
3: .comment S T A R  W A R S ~ END TITLLE ~ (for GS) by Tsugy.
4:
5: (i)
6: (b1)
7:
8: (m1,2000)(amidi1,1)          /   スターウォーズのエンドタイ
```


STAR WARS,
from : STAR WARS
by John Williams

```
  9: (m2,2000)(amidi2,2)          / トルから主節（？）のみ作りま
 10: (m3,2000)(amidi3,3)          / した。すんげー採譜が辛かった
 11: (m4,2000)(amidi4,4)          /  。つづく。
 12: (m5,2000)(amidi5,5)
 13: (m6,2000)(amidi6,6)          /   ○ャラのオーマイアニメは
 14: (m7,2000)(amidi7,7)          / おもろい（ろーかる）。
 15: (m8,2000)(amidi8,8)          / つづく。
 16: (m9,2000)(amidi9,9)
 17: (m10,2000)(amidi11,10)
 18: (m11,2000)(amidi12,11)
 19: (m12,2000)(amidi13,12)
 20: (m13,2000)(amidi10,13)
 21: (m14,2000)(amidi10,14)
 22:
 23: .roland_exclusive $10,$42={$40,$00,$7f,$00}
 24:
 25: /.sc55_v_reserve $10={2,2,2,2,2,2,2,2,1, 2,1,4}
 26:
 27: (t1)  @i$41,$10,$42 x$40,$01,$30,3 x$40,$01,$38,4
 28: (t2)  @i$41,$10,$42
 29: (t3)  @i$41,$10,$42
 30: (t4)  @i$41,$10,$42
 31: (t5)  @i$41,$10,$42
 32: (t6)  @i$41,$10,$42
 33: (t7)  @i$41,$10,$42
 34: (t8)  @i$41,$10,$42
 35: (t9)  @i$41,$10,$42
 36: (t10) @i$41,$10,$42
 37: (t11) @i$41,$10,$42
 38: (t12) @i$41,$10,$42
 39: (t13) @i$41,$10,$42
 40:
 41: / Brass.---------------------------------------------
 42:
 43: (t1) r4 @57@u120@v120q7@p85o418 @e110,40 r*49
 44: (t1) t115 q7c4r{cc}c4r{cc} q6c4c4q7ccq6{ccc}4  [do]
 45: (t1) |:q7c4r{>gg<}c4r{>gg<} q6c4q5>g4<q7ccq6{>ggg<}4:|
 46: (t1) e-2.{e-e-e-}4t117d2.{ddd}4q7t119ddq6t122{ddd}4q4c4
 47: (t1) @62o4@u100q7c4
 48: (t1) f2g4.q8{a-b-} q7a-2q8c4q7c4
 49: (t1) f4.g&a-q8{fa-}q7b+4 b-2^rc4
 50: (t1) f4.ga-.g16<c.>a-16 <f2>f4q8{a-gf}4
 51: (t1) q7<c4.>a-q8c4c6q7c12 f2.c4
 52: (t1) f2g4.q8{a-b-} q7a-2q8c4q7c4
 53: (t1) f4.g&q6a-fq7{a-fb+}4 b-2^rc4
 54: (t1) f4.ga-.g16<c.>a-16 <f2>f4q8{a-gf}4
 55: (t1) q7<c4.>q8{a-f}c4c6q7c12
 56: (t1) |:q7f4q6r{ff}:|f4f4ff{fff}4
 57: (t1) |:q7a4q6r{aa}:|t-7<c4t-8c4t-7ct-4ct-3{ccc}4>
 58: (t1) t100 @58o5@u127@v20q8p3 r*2688
 59: (t1) |:30d-64&~3:|@v127|:18d-64&:||:15d-64&_6:|d-64
 60: (t1) t96r4 t91r4 t87r4 t80r4
 61: (t1) t110 @58@p85 @u98@v120 @k-3
 62: (t1) o4q7 d-2c2 d-4g-a-b-4<c4> a-2d-4e-4 d-2a-4g-4
 63: (t1) a-4<c4>b-2 a-4g-8.f16g-4f8.e-16
 64: (t1) @62q8 @k0~7r4@u99f8.d-16@u117e-4.@u127q7d- d-_r
 65: (t1) t115 @57@u120o4 r2.
 66:
 67: (t2) r4 @57@u120@v120q7@p85o418 @e110,40 r*48
 68: (t2) r1r1 [do]
 69: (t2) |:q7e4r{ee}e4r{ee} q6e4q5d4q7eeq6{ccc}4:|
 70: (t2) g2.{ggg}4g2.{ggg}4q7g4q6{ggg}4q4e4c4
 71: (t2) f2g4.q8{a-b-} q7a-2q8c4q7c4
 72: (t2) f4.g&a-q8{fa-}q7b+4 b-2^rc4
 73: (t2) f4.ga-.g16<c.>a-16 <f2>f4q8{a-gf}4
 74: (t2) q7<c4.>a-q8c4c6q7c12 f2.c4
 75: (t2) f2g4.q8{a-b-} q7a-2q8c4q7c4
 76: (t2) f4.g&q6a-fq7{a-fb+}4 b-2^rc4
 77: (t2) f4.ga-.g16<c.>a-16 <f2>f4q8{a-gf}4
 78: (t2) q7<c4.>q8{a-f}c4c6q7c12
 79: (t2) |:q7f4.q6{ff}:|f4f4ff{fff}4
 80: (t2) |:q7a4.q6{aa}:|<c4c4cc{ccc}4>
 81: (t2) r*1440 o3@u106q6ra-q7a-.a-16
 82: (t2) q7b-4.b-<g-fe-d- q8{d-e-f}4e->b-q6<c4>a-8.q7a-16
 83: (t2) q7b-4.b-<g-fe-d- a-8.e-16e-2>q6a-8.q7a-16 @u111
 84: (t2) q7b-4.b-<g-fe-d- q8{d-e-f}4e->b-q6<c4>a-8.q7a-16
 85: (t2) q8<<d->barg-ed-r a-1
 86: (t2) o4@u103q7 @k2~7 d-2c2 d-4@u92g-a-@u97b-4@u100<c4>
 87: (t2) @u92a-2@u97d-4e-4 d-2@u+3a-4g-4@u92a-4@u+8<c4>b-2
 88: (t2) @u-7a-4g-8.f16g-4f8.e-16 a-4f8.d-16@u+9e-4.d- d-r
 89: (t2) @57@u120@v120o4 @k0 r2.
 90:
 91: (t3) r4 @57@u120@v120q7@p85o418 @e110,40 r*48
 92: (t3) r1r1 [do] r1r1
 93: (t3) q7g4r{gg}g4r{gg} q6g4g4q7ggq6{ggg}4
 94: (t3) q7b-2^rq6{b-b-b-}4b2.{bbb}4<q7c4q6{ccc}4q4c4r4
 95: (t3) @58o3@u127@v127q6
 96: (t3) r*1488c4
 97: (t3) f2g4.q8{a-b-} q7a-2c4c4
 98: (t3) f4.g&q6a-fq7{a-fb+}4 b-2^rc4
 99: (t3) f4.ga-.g16<c.>a-16 <f2>f4q8{a-gf}4
100: (t3) q7<c4.>{a-f}q6c4q7c6c12_7
101: (t3) |:q7f4.q6{ff}:|f4f4ff{fff}4
102: (t3) |:q7a4.q6{aa}:|<c4c4>@57@u120aa{aaa}4
103: (t3) o4@u100q7 r*2688 aa-g-red->ar<@u127q6f4{fff}4f4f4
104: (t3) @58o3@u80q7p3
105: (t3) a-1 g-2a-4g-4 a-2b-2 g-1 a-2b-2 a-2.f4 a-1a-r
106: (t3) @57@u120@p85o4 r2.
107:
108: / Strings.-------------------------------------------
109:
110: (t4) r4 @49@u120@v127q3@p48o418 @e110,40 r*48
111: (t4) r1r1 [do]
112: (t4) |:r4c4r2r1:| o3q7_42112
113: (t4) ge-cae-cb-ge-<c>b-g <d>bg<e-d>b<fdcgdc
114: (t4) gdc~q318o4{ccc}4c4r4
115: (t4) o5@u90
116: (t4) |:r4crrcrc r4crrc{ccc}4 r4crrcrc >r4{b-b-b-}4b-b-<cc
117: (t4)   r4crrcrc r4d-d-d-d-{d-d-d-}4
118: (t4)   r4crrcrc |r4cccc{ccc}4:|r1r1r4c4r2r1
119: (t4) @50o4@u90q8
120: (t4) d-2^e-c.>a-16< d-e-g-2fe- a-4.<c>fd-b-g-
121: (t4) <d-2.c>b- a-4.f<e-d->a-f a-4.{b-a-}ge-b-8.a-16
122: (t4) a-e-<c8.>b-16b-f<{e-d-c}4 d-4.{e-d-}crr4
123: (t4) @49o4@u67 r*768
124: (t4) q8b-4.b-<g-fe-q7d- q8{d-e-f}4e->b-<c4>q7a-8.a-16
125: (t4) @u80 <aa-g-red->ar< f1 q8o4
126: (t4) a-1 g-2a-4g-4 a-2b-2 g-1 a-2b-2 a-2.f4 a-1
127: (t4) @u110o5q3 d-4{d-rd-}4d-r @u120o4 r4
128:
129: (t5) r4 @49@u120@v127q3@p48o418 @e110,40 @k3 r*48
130: (t5) r1r1 [do]
131: (t5) |:r4e4r2r1:| r1r1r4{eee}4e4r4
132: (t5) o4@u90
133: (t5) |:r4frrfrf r4frrf{ggg}4 r4frrfrf r4{fff}4ffgg
134: (t5)   r4frrfrf r4ffff{fff}4 r4frrfrf |r4ffff{fff}4:|
135: (t5) r1r1r4c4r2r1
136: (t5) o4@u60q8
137: (t5) d-2^e-c.>a-16< d-e-g-2fe- a-4.<c>fd-b-g-
138: (t5) <d-2.c>b- a-4.f<e-d->a-f a-4.{b-a-}ge-b-8.a-16
139: (t5) a-e-<c8.>b-16b-f<{e-d-c}4 d-4.{e-d-}crr4>
140: (t5) @u70{>b-<cd-fe-d-cd-}4'd->b-2''b-<d-4''d-f2'
141: (t5) @u78 f&{e-&d-}g-&{f&e-}a-&{g-&f}'b-<d-2''d-b-4'
142: (t5) 'e-a-'120{<cd->}'a-<e-4' @u85'd-b-4'{<d-c>b-}4
143: (t5) <d-r{fe-d-}4 a-r{fe-d-}4frr4 <d->barg-ed-r a-1
144: (t5) @u120
145: (t5) d-2^e-c.>a-16< d-e-g-2fe- a-4.<c>fd-b-g-
146: (t5) <d-2.c>b- a-4.f<e-d->a-f <a-4g-8.f16g-4f8.e-16
147: (t5) a-4f8.d-16e-4.q7d- d-r o4q3 r2.
148:
149: / Bass.----------------------------------------------
150:
151: (t6) r4 @59@u100@v127q4@p55o214 @e110,40 r*48
152: (t6) r1r1 [do]
153: (t6) |:rcrrr1:| r1r1r{ccc}cr
154: (t6) @u80
155: (t6) |:rfr8f8r8f8 rfr8f8{ccc}
156: (t6)   rfr8f8r8f8 r{b-b-b-}b-8b-8<c8c8>
157: (t6)   rfr8f8r8f8 rd-8d-8d-8d-8{d-d-d-}
158: (t6)   rfr8c8r8c8 |rf8f8f8f8{fff}:| rfrrr1q8rf*336
159: (t6) @40o2@u60q7
160: (t6) d-2>a-2g-2a-g-a-2{b-<c}d->g-2a-g-a-2b-2
161: (t6) a-2<e-2>a-a-b-b- <d-{c>b-}a-2 b-*384 b-*384
162: (t6) |:4{b-rb-}b-:||:{f+rf+}f+:| @u70r1
163: (t6) <d-c>a-g- g-b-a-g- a-<c>b-<d-> g-g-a-g-
164: (t6) a-a-b-b- a-<ce->b- a-a-<e-e-
165: (t6) @59@u80q4 d-{d-&rd-}d- @u100 r4
166:
167: / Oboe.----------------------------------------------
168:
169: (t7) r4 @69@u70@v127q3@p52o418 @e110,40 r*48
170: (t7) r1r1 [do]
171: (t7) |:r4c4r2r1:| r1r1r4{ccc}4c4r4
172: (t7) |:r4crrcrc r4crrc{ccc}4 r4crrcrc >r4{b-b-b-}4b-b-<cc
173: (t7)   r4crrcrc r4d-d-d-d-{d-d-d-}4
174: (t7)   r4crrcrc |r4cccc{ccc}4:|r1r1r4c4r2r1
175: (t7) @u50o3q7 r*2304
176: (t7) |:4{b-&rb-}4b-4:||:{f+&rf+}4f+4:| @u60 r1
177: (t7) a-1 g-2a-4g-4 a-2b-2 g-1 a-2b-2 a-2.f4 a-1a-8r8
178: (t7) @u70o4q3 r2.
179:
180: (t8) r4 @69@u70@v127q3@p52o418 @e110,40 r*48
181: (t8) r1r1 [do]
182: (t8) |:r4g4r2r1:| r1r1r4{ggg}4g4r4
183: (t8) |:r4frrfrf r4frrf{ggg}4 r4frrfrf r4{fff}4ffgg
184: (t8)   r4frrfrf r4ffff{fff}4 r4frrfrf |r4ffff{fff}4:|
185: (t8) r1r1r4c4r2r1 r*4608
186:
187: / Flute.---------------------------------------------
188:
189: (t9) r4 @73@u80@v127q3@p52o518 @e110,40 r*48
190: (t9) r1r1 [do]
191: (t9) |:r4c4r2r1:| r1r1r4{ccc}4g4r4
192: (t9) @u100
193: (t9) |:r4crrcrc r4crrc{ccc}4 r4crrcrc >r4{b-b-b-}4b-b-<cc
194: (t9)   r4crrcrc r4d-d-d-d-{d-d-d-}4
195: (t9)   r4crrcrc |r4cccc{ccc}4:|r1r1r4c4r2r1
196: (t9) r*2304 @u65o5
197: (t9) q8b-4.b-<g-fe-q7d- q8{d-e-f}4e->b-<c4>q7a-8.a-16
198: (t9) q8<d->barg-ed-r a-1 @u80q3 r*1536
199:
200: (t10) r4 @73@u80@v127q3@p52o418 @e110,40 r*48
201: (t10) r1r1 [do]
202: (t10) |:r4g4r2r1:| r1r1r4{ggg}4g4r4
203: (t10) @u100
204: (t10) |:r4frrfrf r4frrf{ggg}4 r4frrfrf r4{fff}4ffgg
205: (t10)   r4frrfrf r4ffff{fff}4 r4frrfrf |r4ffff{fff}4:|
206: (t10) r1r1r4c4r2r1 r*4608
207:
208: / O.Hit.---------------------------------------------
209:
210: (t11) r4 @56@u90@v127q4@p43o418 @e110,40 @k-2 r*48
211: (t11) r1r1 [do]
212: (t11) |:r4'ce4'r2r1:| r1r1r4{ccc}4c4r4
213: (t11) |:r4f4rfrf r4f4rf{ccc}4
214: (t11)   r4f4rfrf r4{b-b-b-}4b-b-<cc>
215: (t11)   r4f4rfrf r4d-d-d-d-{d-d-d-}4
216: (t11)   r4f4rcrc |r4ffff{fff}4:||:r4f4r2r1:|
217: (t11) @49o4@u50@p70q8
218: (t11) 'a-<d-2''a-<e-2''g-<d-2''a-<c4'&'g-<d-4'
219: (t11) 'a-<e-2.''a-<d-4''g-b-2''a-<c4'&'g-<d-4'
220: (t11) 'a-<e-2''b-<d-2''a-<d-2''b-<e-2'
221: (t11) q7@u60 'a-<e-4''a-<e-4''b-<d-4''b-<d-4''a-<d-4''g-<c4'
222: (t11) 'a-<c''a-<c''a-<c8.''a-<c16'
223: (t11) r*1536
224: (t11) @56@u90q4@p43o4 r*1344 d-r{d-rd-}4d-rrr
225:
226: / Harp.----------------------------------------------
227:
228: (t12) r4 @47@u90@v100q8p3o3132 @e110,20 r*48
229: (t12) r1r1 [do]
230: (t12) r*2688 r4 fa<cfa<cfa> cfa<cfa<cf>> a<cfa<cfa<c>
```

```
231: (t12) r*4992 o3@u73 r2
232: (t12) {fa-<cfa-<cfa-> cfa-<cfa-<cf>> a-<cfa-<cfa-<c>}2
233: (t12) r*1536 @u90
234:
235: / Prc.--------------------------------------------------
236:
237: (t13) r4 @49@u90@v127q8p3o2l8@r1 @e110,40
238: (t13) @y28,43,64 @y28,41,64 @y28,38,64 r4
239: (t13) r1r1 [do] |:r@u80{gg}@u105g4r2r1:|
240: (t13) r1r1 r4@u90{ggg}4@u110g4r4
241: (t13) r*1344 g1 r*1344
242: (t13) |:r@u80{gg}@u105g4r2|r1:|r2_88|:16^4g32:|
243: (t13) r*1536 @u60r2 |:18{ggg}8:|@u98g4f4g2g4
244: (t13) @u127r*576 _62|:8{ggg}8^8:|
245: (t13) g*1200 @u70 |:6{ggg}8:| @u90d4{drd}4d4r4 @v127
246:
247: (t14) r4 @u90@v127q8p3o3l8@r1 @e110,40
248: (t14) @y28,59,64 @y24,59,66 r*47
249: (t14) r1r1 [do] |:r4b4r2r1:| r1r1r4b4b4r4
250: (t14) r*1344 b1 r*1344 |:r4b4r2r1:|
251: (t14) r*3072  @u108 b1 @u90 r*1344
252:
253: /-----------------------------------------------------
254:
255: (t1)        / [loop] / この部分を有効にするとループします
256: (t2)        / [loop]
257: (t3)        / [loop] / 本当はループした後のイントロは、
258: (t4)        / [loop] / Cじゃなくて F なんだけどまぁいいや
259: (t5)        / [loop] / おわり。
260: (t6)        / [loop]
261: (t7)        / [loop]
262: (t8)        / [loop]
263: (t9)        / [loop]
264: (t10)       / [loop]
265: (t11)       / [loop]
266: (t12)       / [loop]
267: (t13)       / [loop]
268: (t14)       / [loop]
269:
270: (p)
```

リスト5　スターウォーズ用カウンタ表示

```
 1:00002761 00000000   2:00002760 00000000   3:00002760 00000000   4:00002760 00000000
 5:00002760 00000000   6:00002760 00000000   7:00002760 00000000   8:00002760 00000000
 9:00002760 00000000  10:00002760 00000000  11:00002760 00000000  12:00002760 00000000
13:00002760 00000000  14:0000275F 00000000
```

リスト6　明日への扉

日本音楽著作権協会(出)許諾第9372279-301号

```
  1: .comment T-SQUARE より「明日への扉」  Presented By MaSa
  2:
  3:
  4:
  5: / G S 対応 [ CM500  C-mode ]
  6:
  7: (i)
  8:
  9: /-----------------------------------------------------------
 10: /Internal
 11: /  Frequuncy  Moduration sound
 12: (m1, 100)(aFm1,1)
 13: (m2,1000)(aFm2,2)                    / Main 1
 14: (m3,1000)(aFm3,3)                    / Main echo3
 15: (m4,1000)(aFm4,4)                    / Main echo1
 16: (m5,1000)(aFm5,5)                    / Main echo2
 17:
 18: / G S
 19: (m10,1500)(aMidi10,10)               / Bass  Drum
 20: (m11,1000)(aMidi10,11)               / Snare Drum & Tom Tom
 21: (m12,1400)(aMidi10,12)               / Cymbal & HH
 22: (m13,2300)(aMidi 1,13)               / Bass
 23: (m14,2900)(aMidi 2,14)               / Strings
 24: (m15,4700)(aMidi 3,15)               / E.Piano RH
 25: (m16,2300)(aMidi 4,16)               / E.Piano LH
 26: (m17,4600)(aMidi 5,17)               / Guitar
 27: (m18,2600)(aMidi 6,18)               / Ewi
 28: (m19,2600)(aMidi 7,19)               / Ewi(Delay)
 29: (m20,1700)(aMidi 8,20)               / Bell
 30:
 31:
 32: / Exclusive -----------------------------------------------
 33:
 34: .roland_exclusive $10,$42={                  / GS reset
 35:                  $40,$00,$7F,$00}
 36:
 37: .roland_exclusive $10,$42={
 38:                  $40,$10,$0A
 39:                  $01}
 40:
 41: .sc55_reverb ={3,3,2,80,70,70,0}
 42: .sc55_chorus ={3,2,90,8,90,5,20,0}
 43:
 44:
 45: / Sound Set -----------------------------------------------
 46:
 47: / D.Guitar
 48: (@70,     31,  3,  3,  3,  3, 14,  0,  2,  7,  0,  0
 49:           26,  5,  5,  6,  3, 35,  0,  1,  7,  0,  0
 50:           29,  5,  4,  5,  2, 32,  0,  1,  7,  0,  0
 51:           28,  6,  6,  5,  0,  3,  0,  1,  7,  0,  0
 52:            1,  5)
 53:
 54: / Brass 1
 55: (@71,     15, 14,  7,  7,  5, 25,  0,  1,  0,  0,  0
 56:           15, 10,  3,  7, 10, 28,  0,  7,  0,  0,  0
 57:           15,  7,  3,  7,  5, 31,  0,  2,  0,  0,  0
 58:           20,  8,  4,  8,  1,  0,  0,  1,  0,  0,  0
 59:            2,  7)
 60:
 61: / Brass 2
 62: (@72,     31,  8,  8,  5,  0, 24,  1,  5,  7,  0,  0
 63:           25,  4,  5,  4,  2,  5,  1,  4,  7,  0,  0
 64:           27,  4,  5,  4,  2,  5,  1,  3,  7,  0,  0
 65:           29,  4,  5,  4,  2,  5,  1,  2,  7,  0,  0
 66:            5,  7)
 67:
 68:
 69: / MML Data Set --------------------------------------------
 70: (t1)    t120r1
 71: (t2)    r2
 72: (t3)    r2
 73: (t4)    r2
 74: (t5)    r2
 75: (t10)   r2
 76: (t11)   r2
 77: (t12)   r2
 78: (t13)   r2
 79: (t14)   r2
 80: (t15)   r2
 81: (t16)   r2
 82: (t17)   r2
 83: (t18)   r2
 84: (t19)   r2
 85: (t20)   r2
 86:
 87: / D r u m s ---------------------------------
 88: (t10)   o2 @v108 @u104 @p64 q8 l4 @i$41,$10,$42 @e100,20
 89: (t10)   |:|:3crc8c8r:|crc8c8r8c8 |:crc8c8r:|
 90: (t10)   crc8c8|r ccc8c8c:|l8rc rcrcc4c4
 91: (t10)   |:3rcr2rc:| rcr4.c4. c4c4r4.c|:rcr2rc:|rcrcc4r4
 92: (t10)   |:c4rcr4.c | r4.crcrc:| r2c4cc c4rcr4.c rcrcr2
 93: (t10)   |:4rc:| rcrcrc16c16rc c4r4ccr4 c4r2.l4
 94: (t10)   |:6crc8c8r4:| crcr8c8 r2c8c8r r2r8.r16r r8.r16r2
.cr2.
 95: (t10)   |:7crc8c8r:| cccc |:6crc8c8r:|
 96: (t10)   crcc8c8 l16|:rccc:|r4rccc
 97: (t10)   l8c4r4r4.c |:3rcr4r4.c:| ccr4r4.c rcr4.crc
 98: (t10)   r4r4.cr16c16c rc16c16rcc4c.c16 c4r4.c16c16rc
 99: (t10)   r4.crc16c16rc r4.cr16c16crc r4.cr16c16cr4
100: (t10)   cc16c16rcrc16c16rc r4.cr16c16crc r|:7c:| rcrcr2
101: (t10)   c4rcr4.c r4.crcrc c4rcr4.c r4c4r4cc
102: (t10)   c4rcr4.c |:3rc:|r4|:4rc:| ccrcrcc4 |:6c:|r4c4r2.
103: (t10)   l4|:6crc8c8r4:| crc8c8r crc8c8r8c8 |:3crc8c8r:|
104: (t10)   l8c4r4.crc |:c4r4ccr4:| c4r4ccrc rcrcc4r4l4
105: (t10)   ccc8c8r |:3crc8c8r:| crc8c8r8c8 crcc cccc cccc
106: (t10)   l8ccr4ccrc |:5c4r4ccr4:| c4rcc4cc rcrcc.c16rcl4
107: (t10)   |:7crc8c8r:| c|:6c8:|
108: (t10)   ;$cc,10
109: (t10)   l8ccr4ccrc |:c4r4ccr4:| c4rccc16c16rc c4r4ccr4
110: (t10)   c4r4c4c4 cccrc4c4 r16c32c32rc.c16rcc.c16
111:
112:
113: (t11)   o2 @u115 @p64 l4
114: (t11)   r|:30d2:|d.d8d8d8
115: (t11)   |:15d:||:4d16:| r|:13d:|l8rddd
116: (t11)   |:3r2.d4:|r<d32d..>ard4. r2.d4 r4d4.ddd |:8d4:|
117: (t11)   r4d2d<{d>ba} U+5rdd4d4d16d16<d>U-5
118: (t11)   r4|:13d2:|d4 rdd<{d>ba}r2 l16rddddd<ddr>ddr8.aa
119: (t11)   rff8r2r8{<d>ba}8 l8rfffffd16d.
120: (t11)   r4|:28d2:|r4.{<d>ba}r4 d4d4dd16d16d4
121: (t11)   l4r|:29d:||:7d16:|r16 r|:24d:|
122: (t11)   l16d8.dr8dd<dd>bb d4d4r8d8d8d8
123: (t11)   l8|:3r2.d4:| r<d32d16.r>{aaa}afr4 r2.d4 r4d16d.
124: (t11)   d16d.dd |:7d4:|{dddddd}4 r4d2d{<d>ba} rdd4d4d16d
16<d>
125: (t11)   r4|:6d2:|d4.dd4 r4|:22d2:|d4.ddd r4|:11d2:|rd4
126: (t11)   ddd4ddd4l16dddd8<U+4dd8dd>bbaaU-4l4
127: (t11)   r|:15d2:|r
128: (t11)   r|:13d2:|d r8{<d>ba}8rdd8d8
129: (t11)   r|:11d2:|r8d2d8d.dd.d.d8
130:
131:
132: (t12)   o2 @u98 @p64 l8
133: (t12)   <U+3c+U-3>|:6f+:|a+ |:|:15f+:|a+ |:7f+:|a+:|
134: (t12)   f+|:6f+16:|f+f+r<U+2a c+U-2>|:6f+:|a+
135: (t12)   |:3|:7f+:|a+:| <a>|:6f+:|a+ |:7f+:|a+
136: (t12)   |:5f+:|a+f+<U+1c+U-1> |:3f+:|a+f+U+17fff
137: (t12)   fU-13<c+|:14g4:|r4 a4c+4|:12g4:|r>U+13fff @u98
138: (t12)   <a4>f+f+f+a+f+<a> |:7f+:|a+ <c+>|:4f+:|a+f+<c+>
139: (t12)   |:3g+4:|g+<a c+4>rf+f+a+f+<c+4.>f+a+f+U+17fff
140: (t12)   fU-17<ag4gc+g4 gag4g4c+4 c+>|:6f+:|r ru+17fU-17
141: (t12)   <a4>r16a+8.r4<c+|:27b4:|c+4>U+17ffrU-17r<a4.
142: (t12)   r2r8.a8.r r8.c+*108r4 r4>f+f+f+f+ra+<
```

▶「ビバリーヒルズ90210」を見よう！　　前田　万喜雄(18)愛知県

```
   143: (t12)   c+>|:14f+:|a+|:11f+:|f+16f+16f+f+f+a+f+a+|:6f+:|
   144: (t12)   |:7f+:|a+ |:8f+:|116 f+f+f+8r8f+f+f+8f+<a>18ra+
<c+>|:4f+:|a+<a>f+
   145: (t12)   |:5|:7f+:|a+:|a+f+f+a+f+r4<c+ g4g4g4r16b16b16b16
   146: (t12)   c+4|:14g4:|g8a8 c+8a8|:13g4:|r4..a16
   147: (t12)   c+4a4|:13g4:|c+4 a4|:10g4:|r4 a8b4c+2r8
   148: (t12)   c+4>f+f+f+a+f+<a>f+f+f+<c+>f+<c+ac+a>|:4f+:|a+f+
   149: (t12)   <c+4r2.ac+4.a4>a+f+<c+4.>raraU+17fffU-17<c+g4ga8
   150: (t12)   g4gc+g4g4a4 c+>|:6f+:|r rU+17fU-17<c+4r16a8.r4
   151: (t12)   c+|:14b4:|c+c+4.|:15b4:|ac+bc+|:14b4:|ba|:11b4:|
   152: (t12)   c+4bbbgaaa c+4a4b|:4b4:|bbb rbbb rbbb bbbb bbab
   153: (t12)   c+bbbrbbb |:7b:|c+4U+17>ffU-17<c+4>U+17ffU-17<a4
   154: (t12)   r2..c+c+a4> |:51f+:|<c+4>|:6f+:|<a16a16
   155: (t12)   c+b|:27>f+<b:|>f+r<c+>f+<c+4aa4 c+c+4>
   156: (t12)   f+f+f+f+16<c+16c+>|:48f+:|<ac+>f+<c+4a8.a16
   157:
   158:
   159: / B a s s ---------------------------------
   160: (t13)   o2 @33 @v110 @u111 18 @k+2 K0 @p64 @q1 @e12
   161: (t13)   >|:|:8b-:| aaaa<dddd> |:8g:| ffffaaaa|:8b-:|
   162: (t13)   aaaa<dddd> gggg<e-e-e-|e- |:8c:|>:|<c&|:8c:|
   163: (t13)   (c8.<c)>r*660U-4>g<cc> reeffr4f
   164: (t13)   fggr4aag gr4ffggr garb-4f<c4>U+4
   165: (t13)   b-4rb-2a4.ra4.a4 g4rg2f4.rf4.<cd e-4re-2d&d
   166: (t13)   dddd>ad4 rg4gra4a rb-4b-r<c4c |:8d:| rcc4cccc
   167: (t13)   >|:|:8b-:|aaaa<dddd>||:8g:|ffffaaaa:|gggg<e-e-e-
   168: (t13)   c4ccr4c4.116rg<ecd>b-<c>ab-gdc8.b-gafc>f*30018ga
   169: (t13)   |:|:7b-:|b-16b-16 aaaa<dddd>| |:8g:|ffffaaaa:|
   170: (t13)   gggg<|:4e-:| |:7c:|>a|:8b-:| aaaa<dddd> |:8g:|
   171: (t13)   ffffaaaa |:7b-:|b-16b-16 aaaa<dddd>
   172: (t13)   gggg<e->b-<e-c4cccc>g<cc&(c8.<c)>r1618
   173: (t13)   r*1416>g<c4&(c>e)eeffr4f fggr4aag gr4ffggrgarb-4
   174: (t13)   f<c4&(c>e)eeffr4f fggr4aag gr4ffggr garb-4g<c4>
   175: (t13)   b-4rb-2a a4ra4<ae>a g4rg2f4.rf4.<cde-4re-2d4dddd
   176: (t13)   >ad4 rg4gra4a rb-4b-r<c4c |:8d:| rcc4cccc>
   177: (t13)   |:|:|:8b-:| aaaa<dddd>||:8g:|ffffaaaa:|
   178: (t13)   g|ggg<|:4e-:||:7c:|>a:| ab-<de->b-<dc4ccccc>ga
   179: (t13)     |:|:8b-:| aaaa<dddd>||:8g:|ffffaaa<c>:|
   180: (t13)   U+4ggg<de->b-<dc4c116c<c>c8c8c<c>c<c>>aa18U-4
   181: (t13)   |:|:|:7b-:|b-16b-16aaaa<ddd|d16d16>|:8g:|ffff
   182: (t13)   aaaa:|<d> ggg<de->b-<dc4cccc|cc16<c16>>a:|<(cb-)
   183: (t13)   b-b-b-b-afdc>b-g aa<cc+d>ad4 |:8g:|fff(fa)af16f1
6a<c>
   184: (t13)   |:8b-:|aaaa<dddd>ggg<de->b-<dc4cc>g16c16cg<c>a16
a16
   185:
   186:
   187: / S t r i n g s---------------------------
   188: (t14)   o3 @49 @v97 @u97 12 @k0 @q1 @e70 @p64 K0
   189:. (t14)  |:'a1<d' 'g<ce''a<cf' |'f1b-<d'  'ga<c''g<cf':|
   190: (t14)   'b-<dg''b-<dg' '<c1eg'
   191: (t14)   |:'a1<d' 'g<ce''a<cf' |'b-*120<dg'<'f8a''g8b-'
   192: (t14)   'f4.a''e4g''d4f''c4e'>:|'b-2<dg'14'b-.e-g'<'c*72
eg''fa''gb-''a<c'
   193: (t14)   r*1536>
   194: (t14)   'g2..<ce''g*216<ce' 'f2..b-<d''f*216a<c'
   195: (t14)   18'g2..b-<d''a2<df''a4<ce''a4.<cf' r'g4.b-<d'r
   196: (t14)   'a4.<ce'r'a4.<df'r'g4.<ce' 'g1<ce'U-5r@q5'b-<df'
   197: (t14)   @q1'b-4<df'|:4'b-<df':|U+5
   198: (t14)   12|:'a1<d' 'g<ce''a<cf'|'f1b-<d' 'ga<c''g<cf':|
   199: (t14)   'b-<dg''b-4.<dg''b-4<df'|:'b-8<df':|r4'b-4.<dfa'
   200: (t14)   r*576
   201: (t14)     'a1<d' 'g<ce''a<cf''f1b-<d' 'ga<c''g<cf'
   202: (t14)   |:'a1<d' 'g<ce''a<cf'|:'b-<dg':||'<c1eg''a1<d'
   203: (t14)   'g<ce''a<cf' 'b-1dg'< 'f4a''e4g''d4f''c4e'>:|<
   204: (t14)   'c4eg''f4a''g4b-''a4<c'
   205: (t14)   r*3072>
   206: (t14)   'g..<ce''g*216<ce''f..b-<d''f*216a<c''g2..b-<d'
   207: (t14)   18'a2<df''a4<ce''a4.<cf' r'g4.b-<d'r'a4.<ce' r
   208: (t14)   'a4.<df'r'g4.<ce' 'g1<ce' U-5r@q5'b-<df'
   209: (t14)   @q1'b-4<df'|:4'b-<df':|U+5
   210: (t14)   12|:'a1<d' 'g<ce''a<cf'|'f1b-<d' 'ga<c''g<cf':|
   211: (t14)   'b-<dg''b-4.<dg''<c*216eg' 'a1<d' 'g<ce''a<cf'
   212: (t14)   'f1b-<d''ag<c''g<cf''f1a<d''g<ce''a<cf''b-<dg'
   213: (t14)   'b-4.<dg''<c*216eg'|:'a1<d''g<ce''a<cf'|
   214: (t14)   'b-*120<dg'<'f8a''g8b-''f4.a''e4g''d4f''c4e'>:|
   215: (t14)   'b-<dg''b-4.<dg'<'c4.eg''f4a''g4b-''a4<c'
   216: (t14)   'f1a<c' 'eg<c''fa<c' 'd4.gb-''d4gb-''f8a''g8b-''
f4.a''e4g''d4f''c4e'>
   217: (t14)   'a1<d' 'g<ce''a<cf' 'b-<dg''b-4.<dg'<'c4.eg''f4a
''g4b-''a4<c'
   218: (t14)   'f1a<c' 'ce''cf' 'd*120g''f8a''g8b-''f4.a''e4g''
d4f''c4e'>
   219: (t14)   'a1<d' 'g<ce''<cdf' 'b-..<dg'<'c4.eg''f4a''g4b-'
'a4<c'
   220: (t14)   'a1<c'>'g<ce''a<cf' 'b-4.<dg'<'g4b-''f8a''g8b-''
f4.a''e4g''d4f''c4e'>
   221: (t14)   'a1<d' 'g<ce''a<cf' 'b-..<dg'<'c4.eg''f4a''g4b-'
'a4<c'
   222:
   223:
   224: / S t r i n g s (FM)-----------------------
   225: (t2)    @70 @v124 @u124 p2 @q1 @k+3 o3 18 r*3072 @h12 @s
4 @m20
   226: (t2)    >|:rggaar4a abbr4<cc>bbr4aabbr b<crd4>b<|cc>:|<e
4 @m<
   227: (t2)    r*1728 @71p2 rdd4dddd
   228: (t2)    r*1464 @72p2 a4. r1r.>g*348
   229: (t2)    r*3072 @71 p2
   230: (t2)    >|:4rggaar4a abbr4<cc>b br4aabbr b<crd4>b<cc>:|<
<
   231: (t2)    r*1728 @71p2 rdd4dddd
   232: (t2)    r*9216
   233:
   234:
   235: (t3)    @70 @v124 @u124 p3 @q1 @k-1 o3 18 r*3072 @h12 @s
4 @m20
   236: (t3)    |:reeffr4f fggr4ggg gr4ffggr garb-4g|gg:|<c4> @m
   237: (t3)    r*1728 @71p3 rb-b-4b-b-b-b-
   238: (t3)    r*1464 @72p3<f4. r1r.>e*348
   239: (t3)    r*3072 @70 p3
   240: (t3)    |:4reeffr4f fggr4ggg gr4ffggr garb-4ggg:|
   241: (t3)    r*1728 @71p3 rb-b-4b-b-b-b-
   242: (t3)    r*9216
   243:
   244:
   245: (t4)    @70 @v124 @u124 p3 @q1 @k+1 o3 18 r*3072 @h12 @s
4 @m20
   246: (t4)    |:rccccr4c cddr4eed dr4ccddr derf4d|cc:|g4 @m
   247: (t4)    r*1728 @71p3 rgg4gggg
   248: (t4)    r*1464 @72p3<d4. r1r.>c*348
   249: (t4)    r*3072 @70 p3
   250: (t4)    |:4rccccr4c cddr4eed dr4ccddr derf4dcc:|
   251: (t4)    r*1728 @71p3 rgg4gggg
   252: (t4)    r*9216
   253:
   254:
   255: (t5)    @70 @v124 @u124 p1 @q1 @k-3 o2 18 r*3072 @h12 @s
4 @m20
   256: (t5)    |:reeffr4f fggr4aag gr4ffggr garb-4g|<cc>:|<c4 @
m
   257: (t5)    r*1728 @71p1 rcc4cccc
   258: (t5)    r*1464 @72p1b-4. r1r.>f*348
   259: (t5)    r*3072 @70 p1
   260: (t5)    |:4reeffr4f fggr4aag gr4ffggr garb-4g<cc>:|<
   261: (t5)    r*1728 @71p1 rcc4cccc
   262: (t5)    r*9216
   263:
   264:
   265: / K e y B o a r d--------------------------
   266: (t15)   o3 @1 @v105 @u104 12 @k-1 @p58 @q1 @e40
   267: (t15)   |:|:'a<d'r8'a4.<d''g<ce''a<cf''fb-<d'|r8'f4.b-<d
''ea<c''g<cf':|
   268: (t15)   |'b-<df'<18|:5'ceg':|U+7'ce'dcU-7>12:|'b-4.<df''
b-8<df'&'b-1<df'
   269: (t15)   18@4@e20¯5U+5|:r|:'eg<c':||:'fa<c':|r4|:'fa<c':|
   270: (t15)   |:|:'gb<d':||r4|:'g<ce':||:'gb<d':|r4|:'fa<c':|
   271: (t15)   :|r'gb<d''a<ce'r'b-4<df''gb<d'||:'g<ce':|:|'g4<c
e'
   272: (t15)   @1@e40_5U-5
   273: (t15)   'g4.<ce''g2<ce''g2<ce'<<'e<c''cg'>'g4.<e'>
   274: (t15)   'f4.b-<d''f2b-<d''f2a<c' <g32'g16.<c''g<c''f4.<c
'>
   275: (t15)   'g4.b-<d''g2b-<d''f4a<d'|:'fa<d':|'f4a<e''a4.<ce
'
   276: (t15)   r'g4.b-<d'r'a4.<ce' r'a4.<df'r'<c4.eg' |:8'g<ce'
:|
   277: (t15)   @e15¯5U+6r@q4'b-<df'@q1'b-4<df'|:4'b-<df':|_5U-6
   278: (t15)   |:'a2<d'r'a4.<d''g2<ce''a2<cf''f2b-<d'|r'f4.b-<d
'
   279: (t15)   'e2a<c''g2<cf':|'b-4.df''b-4<df'|:'b-<df':|r4'b-
4.<dfa'
   280: (t15)   ¯5U+5116@4<<rgecd>b-<c>ab-gdc8.b-g afc>@1'b-*348
ge'<
   281: (t15)   18@v114@u115@e60
   282: (t15)   r4.a<defg4c<c>e32f8..ef>b32<'c8..f'>'b-4<f'<
   283: (t15)   'c16f'd16>b32<'c16.f'>fgg64'a*69<c'116rb-afc8>b-
   284: (t15)   a8<cfa'd8.g'f8.d4..fga f64'c*33g'<c8.>f4.(rga)4
   285: (t15)   a32b-16.ab-8<c8d8gab-agdg e>g<(cef)8ec>gfd64
   286: (t15)   'e*21g''dg''c8g'<cde @q0U+3>'a8<df'<f>'g8<ce'<e>
   287: (t15)   'f8.a<d''e8g<c''eg<c''f8a<d'<'c8fa'@q1'c4eg'U+2
   288: (t15)   'c8e<c'U-2'c4.f'rU-3fga b-8(df<c)8>b-afdc>b-a<c>
b-afd
   289: (t15)   c8>a<cf8ga<c>fga<cfef d64e32.rd>a8gfedf8d8fa<d
   290: (t15)   dec>g8<c8gf4ref818U+2'f<d''e<c''db-''c4a'>'b-<g'
   291: (t15)   'a<f'<'c4g'>'g<e''fa<d''e*120<c'U-2
   292: (t15)   @4@e20@v110@u10918>
   293: (t15)   |:4r|:'eg<c':||:'fa<c':|r4|:'fa<c':||:'gb<d':|r4
   294: (t15)   |:'g<ce':||:'gb<d':|r4|:'fa<c':||:'gb<d':|r'gb<d
'
   295: (t15)   'a<ce'r'b-4<df''gb-<d'|:'g<ce':|:|
   296: (t15)   @1@e40@v105@u104
   297: (t15)   'g4.<ce'|:'g2<ce':|<<'e8<c''c8g'>'g4.<e'>'f4.b-<
d'
   298: (t15)   'f2b-<d''f2a<c'<g32'a16.<c''g8<c''f4.<c'>'g4.b-<
d'
   299: (t15)   'g2b-<d''f4a<d'|:'fa<d':|'f4a<e''a4.<ce'r'g4.b-<
d'
   300: (t15)   r'a4.<ce'r'a4.<df'r'<c4.eg'|:8'g<ce':|@e15¯5U+6r
   301: (t15)   @q4'b-<df'@q1'b-4<df'|:4'b-<df':|_5U-6
   302: (t15)   |:'a2<d'r'a4.<d''g2<ce''a2<cf''f2b-<d'|r'f4.b-<d
'
   303: (t15)   'e2a<c''g2<cf':| 'b-2<df'<|:3'ceg':|'c4eg''dg''c
4g'>
   304: (t15)   |:'a2<d'r'a4.<d''g2<ce''a2<cf'|'g4.b-<d'
   305: (t15)   U+5<b-4a4b-U-5 'c4fa'g4'c4f'>'f4<c':|
   306: (t15)   'g2b-<d''b-4.<df'<'c4eg'|:'ceg':|'c4eg''c4.eg'>
   307: (t15)   'f2a<d'r'f4.a<d'14|:'a<ce':||:'a<cf':|'b-.<dg'<
   308: (t15)   'dfb-''f8a''g8b-''c.fa''cg''cf'>'f<c''f.a<d'
   309: (t15)   'fa<d''f.a<d' |:'g<ce':||:'a<cf':| U+3'g2b-<d'
   310: (t15)   'b-.<df'<|:3'ceg':|'c.eg'U-3>
   311: (t15)   |:'f.a<d''fa<d''f.a<d' 'g<ce''<ceg''a<cf'<'fa<c'
   312: (t15)   'd.fb-''dfb-''c8fa''d8fb-''c.fa''cg''cf'>'f<c'
   313: (t15)   'f.a<d'|'fa<d''b-<df'<a8 'eg<c'<c>>|:'a<cf':|
   314: (t15)   'b-2<dg''b-.<dg'<|:3'ceg':|'c.eg'>:|<|:'fa<d':|
   315: (t15)   'f8a<d''g<ce''eg<c''a<cf''fa<d''>b-2<dg''e-.gb-'
   316: (t15)   'c.eg''cea''dfb-''eg<c' 'd.fa'|:'dfa':|'d8fa'
   317: (t15)   'ceg'c'c2f''>g.b-<d''dgb-''f8a''g8b-''c.fa''cg'
   318: (t15)   'cf'>'f<c''f.a<d'<'dfa''d.fa''e2g<c''c2e''d2fb-'
   319: (t15)   'e-.gb-''ceg''c8eg''cea''dfb-''eg<c'
   320:
   321:
   322: (t16)   o2 @1 @v105 @u104 12 @k+1 @p70 @q1 @e40
   323: (t16)   |:|:'b-<f'r8'b-4.<f'a<d>'g<d'|r8'g4.<d'fa:||<e-c
1>:|<e-4.c8&c1
   324: (t16)   18@4@e20¯5U+5> |:r|:'e<c':||:'f<c':|r4|:'f<c':|
   325: (t16)   |:'g<d':|r4|:'a<e':||:'g<d':|r4|:'f<c':||:'g<d':
|
   326: (t16)   r'g<d''a<e'r'b-4<f''g<d'<|cc>:|<c4>
   327: (t16)   @1@e40_5U-5 b-4.b-2a2a4a4. 'g4.<d''g2<d''f2<c'
   328: (t16)   'f4<c''f4.<c' 'e-4.g''e-2g''d2a''d4a''d4.a'r
   329: (t16)   'g4.<d'r'a4.<e' r'b-4.<f'r<'c4.g' |:8d:| @e15
```

▶そろそろ，次期新機種が気になりだしてくる……。　河野　智晃(20)千葉県

```
   330: (t16)   ~5U+6r@q4'cg'@ql'c4g'|:4'cg':|_5U-6>
   331: (t16)   |:'b-2<f'r'b-4.<f' a2<d2> 'g2<d'|r'g4.<d'f2a2:|
   332: (t16)   <e-4.c4ccr4c4.> U+5r*228'f*348<c'U_5
   333: (t16)   r*3072
   334: (t16)   @4@e20@v110@u10918
   335: (t16)   |:4r|:'e<c':||:'f<c':|r4|:'f<c':||:'g<d':|r4
   336: (t16)   |:'a<e':||:'g<d':|r4|:'f<c':||:'g<d':|r 'g<d'
   337: (t16)   'a<e'r'b-4<f''g<d'U+3<cc>U-3:|
   338: (t16)   @1@e40@v105@u104 U+3b-4.b-2a2a4a4.U-3
   339: (t16)   'g4.<d''g2<d''f2<c''f4<c''f4.<c' 'e-4.g''e-2g'
   340: (t16)   'd2a''d4a''d4.a'r'g4.<d'r'a4.<e'r'b-4.<f'r<'c4.g
'|:8d:|
   341: (t16)   @e15~5U+6r@q4'cg'@ql'c4g'|:4'cg':|_5U-6>
   342: (t16)   |:'b-2<f'r'b-4.<f' a2<d2> 'g2<d'|r'g4.<d' f2a2:|
 <e-2 'clg'>
   343: (t16)   |:'b-2<f'r'b-4.<f' a2<d2> 'g2<d'|r'g4.<d''f2<c'<
f4r4>:|<e-4. c&c2rc4.>
   344: (t16)   b-2rb-4. a2<d2> 'g4.<d''g*120<d' 'f2<c'<f4r4>
   345: (t16)   b-4.b-4b-4. a2<d2> U+3g2<e-4. c*216U-3>
   346: (t16)   14b-.b-b-.a2<d2> 'g.<d''g*120<d' 'f2<c'<fr>
   347: (t16)   b-.b-b-a*120<d2> g2<e-.|:3'cg':|'c.g'>
   348: (t16)   b-.b-*120 aa<dd 'a.<f'|:'a<f':|'f*120<c''a<f'r
   349: (t16)   b-.b-b-. 'a2<g''<d2f' 'g2<f'<e-.'c.g'|:3'ce':|>
   350: (t16)   'b-.<f'|:'b-<f':|'b-8f' 'a2<g'<d2> 'g.<df'|:'g<d
f':|'f*120<c''a<f'r
   351: (t16)   b-.b-b-.'a2<e''d<d'<d>'g2<f'<e-.'c2<g''c8g''c2g'
   352:
   353:
   354: / G u i t a r-----------------------------
   355: (t17)   I0o3 @31 @v86 @u88 @p64 18 @k-3 @q9 @e75,20 @G12
   356: (t17)   |:|:|:8'b-<f':||:4'a<e':|<U+6|:4'da':|U-6>|
   357: (t17)   |:8'g<d':||:4'f<c':||:4'a<f':|:||:4'g<d':|<|U+4
   358: (t17)   |:4'e-b-':|U-4|:6'cg':|@b-22,-4096,0'c.g'
   359: (t17)   @b-33r16>:|<U+5|:3'e-b-':|U-5'cg'&|:4'c4g':|
   360: (t17)   18@29U+8@q3 @e20>
   361: (t17)   |:r|:'e<c':||:'f<c':|r4|:'f<c':||:'gb':|
   362: (t17)   r4|:'a<c':||:'gb':|r4|:'f<c':||:'gb':|
   363: (t17)   r 'gb''a<c'r'a4.<d''g4<c':|
   364: (t17)   I0@31 @q2 @e75,10 18 o2
   365: (t17)   'b-2..<b-''a<a'&'al<a' 'g2..<g''f<f'&'f1<f'<<U-4
   366: (t17)   'e-2..b-''d2..a'@b-33,-4779,0'da'r@b-33r>'g4.<d'
   367: (t17)   r'a4.<e' r'b-4.<f'r<'c4.g' U-4@q9|:8'da':| @q2r
   368: (t17)   'cg''c4g'|:4'cg':|>
   369: (t17)   @q9|:|:8'b-<f':| |:4'a<e':|<|U+6|:'da':|
   370: (t17)   U-3@b-33,-4779,0'd8.a'r16@b-33U-3>|:8'g<d':||:4'
f<c':|
   371: (t17)   |:4'a<f':|:|<|:4'da':|> |:4'g<d':|<@q2'e-4.b-'
   372: (t17)   'c4g'|:'cg':|r4'c4.g'~12U+12l16<rgecd>b-<c>ab-gd
c8.b-g
   373: (t17)   afc>U-12_12b-*300<@e100@b-33,-8192,0b-8.@b-33r18
@e75,10>
   374: (t17)   @q9|:8'b-<f':||:4'a<e':|<|:'da':|@b-33,-8192,0'd
8.a'r16@b-33>
   375: (t17)   |:8'g<d':||:4'f<c':||:4'a<f':| |:8'b-<f':||:'a<f
':|
   376: (t17)   @b-33,-8192,0'a4<f'@b-33<|:'da':|@b-33,-8192,0'd
4a'@b-33>
   377: (t17)   |:4'g<d':|<|:4'e-b-':| |:6'cg':|'c4g'>
   378: (t17)   |:|:8'b-<f':| |:4'a<e':|<||:4'da':|>|:8'g<d':|
   379: (t17)   |:4'f<c':||:4'a<f':|:|<|:'da':|@b-33,-8192,0'd4a
'@b-33>
   380: (t17)   |:4'g<d':|<'e-4.b-''c2..<g'
   381: (t17)   I0@30@v113@u113@e100,10q8 @m95@s4@h20 l16 a<cdc
   382: (t17)   @b-33,2940,0d32@b-33'e*114g'dcd>a<c>ara<c>a<c8dc
4..>
   383: (t17)   @b-33,2950,0g32@b-33,-2746,33a*42@b-33gfgfef
   384: (t17)   {efe}8dcdedc> agee-dc>ag&g4r<edc (de)e<c8.>a<cd
   385: (t17)   (d32e)e16.ga<cdc(de)e8.dg8.fefedc(d64e)e32.dc d
   386: (t17)   >a<cd(d32e)e16.g(e-d)dcd(ce)e8ae<|:(c32d)d16.:|
   387: (t17)   c4..>>(cg)g4< {agfgfe}4|:{agfagf}4:|{gfefed}4
   388: (t17)   |:{edcedc}4:|{c>ba<c>ba}4{bagbag}4{bagbag}4
   389: (t17)   {gfegfre}4{gfedc>b}4d8e8 d8f8>a8g4.<<{cdefga}4
   390: (t17)   <{cde}8g8.a8.b8.<c8.d4c8&c2>r8{ega}8
   391: (t17)   |:3{<c>agega}4:|{<c>agage}4{gededc}4(d32e)e16.c*
120
   392: (t17)   I0@31 @v86 @u88 @q2 @e75,20 18 o5 @k-3
   393: (t17)   c2..@mo2'a*216<a' 'g2..<g''f*216<f'<<U+4
   394: (t17)   'e-2..b-''d2..a'@b-33,-4779,0'da'r@b-33r>'g4.<d'
   395: (t17)   r'a4.<e'r'b-4.<f'r<'c4.g'U-4@q9|:8'da':|@q2r'cg'
'c4g'|:4'cg':|>
   396: (t17)   @q9|:|:8'b-<f':| |:4'a<e':|<|U+6|:'da':|U-3
   397: (t17)   @b-33,-4779,0'd8.a'r16@b-33U-3>|:8'g<d':||:4'f<c
':|
   398: (t17)   |:4'a<f':|:|<|:4'da':|>|:4'g<d':|<|:4'e-b-':||:8
'cg':|>
   399: (t17)   |:|:8'b-<f':| |:4'a<e':|<U+6|:'da':|U-3
   400: (t17)   @b-33,-4779,0'd8.a'r16@b-33U-3>||:8'g<d':||:4'f<
c':|
   401: (t17)   |:4'a<f':|:||:4'g<d':|<|:3'e-b-':|'c4.g'@q0~4U+7
   402: (t17)   (d32e)e16.ga16g16e16d16c>b-_4U-7@q9
   403: (t17)   |:|:8'b-<f':| |:4'a<e':|<U+6|:4'da':|U-6>||:8'g<
d':| |:4'f<c':||:4'a<f':|:|
   404: (t17)     |:4'g<d':|<U+5|:3'e-b-':|'c4g'|:7'cg':|>U-5
   405: (t17)   |:3|:|:8'b-<f':| |:4'a<e':|<U+4|:4'da':|U-4>|
   406: (t17)      |:8'g<d':| |:4'f<c':||:4'a<f':|:||:4'g<d':|
   407: (t17)   <|:3'e-b-':|'c4e'|:7'ce':|>:|
   408:
   409:
   410: / E w i-----------------------------------
   411: (t18)   o4@82@v110@u109@p6518@k-2@q2@e10,10@h24@m70@s4
   412: (t18)   |:|:d4fg4edc4(f32g)&g16.ef4|c>ab-4.r<@q14f@q2
   413: (t18)   (f32g)&g16.edc&c1:| gaa32
   414: (t18)   b-8..agf4@q14g@q2a|g4edc4.r4:|g2&grg<cc
   415: (t18)   c>geacr4. r2rl16ga<cdc>al8 <d@q14e@q2>(g32a)&a16
.g&g2 r2rg<cc
   416: (t18)   c>gear2   r2rl16ga<cdc>al8 <d@q14e@q2>(g32a)&a16
.g&g2 r2r<def
   417: (t18)   @b-624,-22,0g&g4@b-22f16e16defe& ed16c16>g4r<
   418: (t18)   ce32f32e16c d4.>b-16a16gab-@b-624,-22,0a&a2
   419: (t18)   @b-22rfga@b-704,-22,0b-&b-4@b-22a16g16fgaf4ag<c>
   420: (t18)   a<def rd>b-g<ec>a<(d32e)&e16. d>b-<dgec>a<
   421: (t18)   @b-1365,-22,0d&d1@b-22 r1>
   422: (t18)   |:d4fg4edc4(f32g)&g16.ef4|c>ab-4.<r@q14f@q2
   423: (t18)   (f32g)&g16.edc&c1:| gaa32
   424: (t18)   b-8..agf4@q14g@q2ag&g2r2
   425: (t18)   r*6648 <def
   426: (t18)   @b-624,-22,0g&g4@b-22f16e16defe& ed16c16>g4r<
   427: (t18)   ce32f32e16c d4.>b-16a16gab-@b-624,-22,0a&a2
   428: (t18)   @b-22rfga@b-704,-22,0b-&b-4@b-22a16g16fgaf4ag<c>
   429: (t18)   a<def rd>b-g<ec>a<(d32e)&e16. d>b-<dgec>a<
   430: (t18)   @b-1365,-22,0d&d4@b-22@q0l16<@b-682,-22,0d@b-22
   431: (t18)   c>agl32fec>aa-gfdl8{c>aa-gfd}4@q2< r1
   432: (t18)   |:|:d4fg4edc4(f32g)&g16.ef4|c>ab-4.<r@q14f@q2
   433: (t18)   (f32g)&g16.edc&c1:| gaa32b-8..agf4@q14g@q2a|
   434: (t18)   g4edc4.r4:| g&g2&g~3U+4ab-<c
   435: (t18)     |:d4fg4edc4(f32g)&g16.ef4|c>ab-2 < @q14f@q2
   436: (t18)   (f32g)&g16.edc&c1:| gaa32b-8..agf4@q14g@q2ag4
   437: (t18)   (g16a)&a8.(a16b-)&b-8.(b-16<c)&c8.>
   438: (t18)   U+1f*120r@q14a@q2<(c16d)&d8.c>g4f4d@q14a@q2
   439: (t18)   (a16b-)&b-8.(g16a)&a8.(e16f)&f8.(c16d)&d8.
   440: (t18)   @b-683,-22,0c8&c2@b-22r2>rg@b-22,-5461,5g@b-22
   441: (t18)   r4a<cega<c(f16g)&g8.fd>a-32g16.fdfgal16b-arfdd-
   442: (t18)   cdfa<c>ab-gafgefcec d8.ar2rdfa g8a@q14g8@q2e8d
   443: (t18)   rc>agfed>f g4r8.<drefgagf+a g8b-8b-r<cdrefgarge
   444: (t18)   f8<@q14d8@q2c8dcr>g8.f8grr4.@q0{cfa}8{b-ab-afd}4
   445: (t18)   >{b-af}8r8@q2r4d8fdf8gfa8b-ab-<cdfgab-<cdefgag32
   446: (t18)   g-32 fgf8d8f8g8rdfarc8e8g8f8d>b-g<ec32>a32f<d>
   447: (t18)   b-32g32e<c>a32f32 db-g32e32caf32d32>b-<ge32c32>a
   448: (t18)   <fd32>b-32g<ec32>a32f<dfedc>bb-a-a<cd>a-g8@q14f8
   449: (t18)   @q2f8a8<c8d8f8g8a8g4a8<e8f2dc
   450: (t18)   >g8fdf8g8g8fdg8g8gdg8g8fdfgab-8afd
   451:
   452:
   453: / E w i(Delay)----------------------------
   454: (t19)   o4@82@v96@u95@p6518@k+2@q2@e53,15r16@h24@m70@s4
   455: (t19)   |:|:d4fg4edc4(f32g)&g16.ef4|c>ab-4.r<@q14f@q2
   456: (t19)   (f32g)&g16.edc&c1:| gaa32
   457: (t19)   b-8..agf4@q14g@q2a|g4edc4.r4:|g2&grg<cc
   458: (t19)   c>geacr4. r2rl16ga<cdc>al8 <d@q14e@q2>(g32a)a16.
g&g2 r2rg<cc
   459: (t19)   c>gear2   r2rl16ga<cdc>al8 <d@q14e@q2>(g32a)a16.
g&g2 r2r<def
   460: (t19)   @b-602,22,0g&g4@b22f16e16defe& ed16c16>g4r<
   461: (t19)   ce32f32e16c d4.>b-16a16gab-@b-602,22,0a&a2
   462: (t19)   @b22rfga@b-682,22,0b-&b-4@b22a16g16fgaf4ag<c>
   463: (t19)   a<def rd>b-g<ec>a<(d32e)&e16. d>b-<dgec>a<
   464: (t19)   @b-1365,22,0d&d1@b22 r1>
   465: (t19)   |:d4fg4edc4(f32g)&g16.ef4|c>ab-4.<r@q14f@q2
   466: (t19)   (f32g)&g16.edc&c1:| gaa32
   467: (t19)   b-8..agf4@q14g@q2ag&g2r2
   468: (t19)   r*6648<def
   469: (t19)   @b-602,22,0g&g4@b22f16e16defe& ed16c16>g4r<
   470: (t19)   ce32f32e16c d4.>b-16a16gab-@b-602,22,0a&a2
   471: (t19)   @b22rfga@b-682,22,0b-&b-4@b22a16g16fgaf4ag<c>
   472: (t19)   a<def rd>b-g<ec>a<(d32e)&e16. d>b-<dgec>a<
   473: (t19)   @b-1365,22,0d&d4@b22@q0l16<@b-682,22,0d@b22
   474: (t19)   c>agl32fec>aa-gfdl8{c>aa-gfd}4@q2< r1
   475: (t19)   |:|:d4fg4edc4(f32g)&g16.ef4|c>ab-4.<r@q14f@q2
   476: (t19)   (f32g)&g16.edc&c1:| gaa32b-8..agf4@q14g@q2a|
   477: (t19)   g4edc4.r4:| g&g2&g~3U+4ab-<c
   478: (t19)     |:d4fg4edc4(f32g)&g16.ef4|c>ab-2 < @q14f@q2
   479: (t19)   (f32g)&g16.edc&c1:| gaa32b-8..agf4@q14g@q2ag4
   480: (t19)   (g16a)&a8.(a16b-)&b-8.(b-16<c)&c8.>
   481: (t19)   U+1f*120r@q14a@q2<(c16d)&d8.c>g4f4d@q14a@q2
   482: (t19)   (a16b-)&b-8.(g16a)&a8.(e16f)&f8.(c16d)&d8.
   483: (t19)   @b-683,22,0c8&c2@b22r2> rg@b22,-5461,5g@b22
   484: (t19)   r4a<cega<c(f16g)&g8.fd>a-32g16.fdfgal16b-arfdd-
   485: (t19)   cdfa<c>ab-gafgefcec d8.ar2rdfa g8a@q14g8@q2e8d
   486: (t19)   rc>agfed>f g4r8.<drefgagf+a g8b-8b-r<cdrefgarge
   487: (t19)   f8<@q14d8@q2c8dcr>g8.f8grr4.@q0{cfa}8{b-ab-afd}4
   488: (t19)   >{b-af}8r8@q2r4d8fdf8gfa8b-ab-<cdfgab-<cdefgag32
   489: (t19)   g-32fg f8d8f8g8rdfarc8e8g8f8d>b-g<ec32>a32f<d>
   490: (t19)   b-32g32e<c>a32f32 db-g32e32caf32d32>b-<ge32c32>a
   491: (t19)   <fd32>b-32g<ec32>a32f<dfedc>bb-a-a<cd>a-g8@q14f8
   492: (t19)   @q2f8a8<c8d8f8g8a8g4a8<e8f2dc
   493: (t19)   >g8fdf8g8g8fdg8g8gdg8g8fdfgab-8afd
   494:
   495:
   496: / B e l l---------------------------------
   497: (t20)   o6 @12 @v94 @u95 @p64 18 @k+1 q8 @e100,30
   498: (t20)   |:|:d4fg4edc4gef4|c>ab-4.r<fgedc&c1:|
   499: (t20)   gab-4agf4ga|g4edc4.r4:|g2&g
   500: (t20)   r*3552
   501: (t20)   |:d4fg4edc4gef4|c>ab-4.<r@q14f@q2gedc&c1:|
   502: (t20)   gab-4agf4gag&g1
   503: (t20)   l16rgecd>b-<c>ab-gdc8.b-g afc>b-*156<<
   504: (t20)   r*3216 q8I8@31@v90@u90@e100,10 @m95@s4@h20l16
   505: (t20)   r32>>  a<cdc
   506: (t20)   @b33,2940,0d32@b33'e*114g'dcd>a<c>ara<c>a<c8dc4.
.>
   507: (t20)   @b33,2950,0g32@b33,-2746,33a*42@b33gfgfef
   508: (t20)   {efe}8dcdedc> agee-dc>ag&g4r<edc (de)e<c8.>a<cd
   509: (t20)   (d32e)e16.ga<cdc(de)e8.dg8.fefedc(d64e)e32.dc d
   510: (t20)   >a<cd(d32e)e16.g(e-d)dcd(ce)e8ae<|:(c32d)d16.:|
   511: (t20)   c4..>>(cg)g4< {agfgfe}4|:{agfagf}4:|{gfefed}4
   512: (t20)   |:{edcedc}4:|{c>ba<c>ba}4{bagbag}4{bagbag}4
   513: (t20)   {gfegfre}4{gfedc>b}4d8e8 d8f8>a8g4.<<{cdefga}4
   514: (t20)   <{cde}8g8.a8.b8.<c8.d4c8&c2>r8{ega}8
   515: (t20)   |:3{<c>agega}4:|{<c>agage}4{gededc}4(d32e)e16.c*
114
   516: (t20)   @mr*1920
   517: (t20)   o6 @12 @v94 @u95 @p64 18 @k+1 q8 @e100,20
   518: (t20)   |:|:d4fg4edc4gef4|c>ab-4.<rfgedc&c1:|
   519: (t20)   gab-4agf4ga|g4edc4.r4:| g&g2&gab-<c
   520: (t20)     |:d4fg4edc4gef4|c>ab-2<fgedc&c1:|
   521: (t20)   gab-4agf4gag4a4b-4<c4>r> r*3072~6U+5
   522: (t20)     |:d4fg4edc4gef4|c>ab-4.<rfgedc&c1:|
   523: (t20)   gab-4agf4gag4edc4.r4
   524:
   525:
   526: (p)
```

▶「LIVE in」にテクノなどが載るとうれしいんですけど（個人的希望）。

松本　健一(21)静岡県

リスト7　明日への扉用カウンタ表示

```
 1:000000C0 00000000   2:000065A0 00000000   3:000065A0 00000000   4:000065A0 00000000
 5:000065A0 00000000  10:000065A0 00000000  11:000065A0 00000000  12:000065A0 00000000
13:000065A0 00000000  14:000065A0 00000000  15:000065A0 00000000  16:000065A0 00000000
17:000065A0 00000000  18:000065A0 00000000  19:000065AC 00000000  20:000065A0 00000000
```

リスト8　夢路より

```
  1: .comment Stephen Foster作曲『Beautiful Dreamer 夢路より』
by kunkun
  2: (i)
  3: (v119,0,58,15,2,0,220,0,0,0,0,3,0,
  4: 28,4,0,5, 1,37,2, 1,7,0,0,
  5: 22,9,1,2, 1,47,2,12,0,0,0,
  6: 29,4,3,6, 1,37,1, 3,3,0,0,
  7: 15,7,0,5,10, 0,2, 1,0,0,1)
  8:
  9:
 10: (m1,8000)
 11: (m2,8000)
 12: (m3,8000)
 13: (m4,8000)
 14: (m5,8000)
 15: (m6,8000)
 16: (m7,8000)
 17: (m8,8000)
 18:
 19: (a1,1)
 20: (a2,2)
 21: (a3,3)
 22: (a4,4)
 23: (a5,5)
 24: (a6,6)
 25: (a7,7)
 26: (a8,8)
 27:
 28: (t1) o4v15q818
 29: T100@119<<dc#d>a4.f#4.ed#eb2.a<gf#f#ed#e>b<c#d2.r64>>R4.
 30: |:T100<<dc#d8>a4.f#4.ed#eb2.a<c#>bbaggf#ef#2.r4.<dc#d>a4.
 31: f#4.ed#eb2.a<c#>bbaggf#ed2.r64>p2v14r64l32ab<c#def#g6418>
 32: p2v14r64<agec#4.b4.baf#d2.<v16T93dc#d>b4.<T50e4T85dT90c#
 33: v14d>ba2.>T95p3v16<<dT100c#d>a4.f#4.ed#eb2.a<c#>bbaggf#e
 34: T90f#4.&f#4.&f#4.>R64p2v14r64<b<c#dd>af#a64gf#ed2.>R16..
 35: |1p2v14l32r64ab<c#def#g64>l8:|
 36: |2T95p3v16<<T92dc#T85d>T80a4.f#4.T78ed#T75eb2.a<gf#f#
 37: ed#e>b<c#d2.r64r32p3v16r16r32d1&d2>>
 38:
 39: (t2) o4v15q818
 40: @119p2v14r64<<dc#d>a4.f#4.ed#eb2.a<gf#f#ed#e>b<c#d2.>>R4.
 41: |:r64<<dc#d>a4.f#4.ed#eb2.a<c#>bbaggf#ef#2.r4.<dc#d>a4.
 42: f#4.ed#eb2.a<c#>bbaggf#ed2.>p3v16a.&a32
 43: p3v16agec#4.b4.baf#8d2.<v16dc#d>b4.<e4dc#v14d>ba2.r64
 44: p2v14r64<<dc#d>a4.f#4.ed#eb2.a<c#>bbaggf#ef#4.&f#4.&f#4.>
 45: p3v16<b<c#dd>af#a64gf#ed2.r64>R16..
 46: |1p3v16a.&a32:|
 47: |2p2v14r64<<dc#d>a4.f#4.ed#eb2.a<gf#f#ed#e>b<c#d2.r64r32
 48: p3v16r32>a1&a2>R64
 49:
 50: (t3) o4v15q818
 51: @119<f#2.r4.>b2.r4.r<baagf#gdea2.r64>R4.
 52: |:<d2.r4.>g2.r4.<c#2.&c#4r>a2.r4.<d2.r4.>g2.r4.<c#2.&c#4r
 53: >f#2.r64p3v16r32b.p3v16<agec#4.b4.baf#d2.<v16dc#d>b4.<e4
 54: dc#v14d>ba2.R64>p3v16<d2.r4.>g2.r4.<c#2.&c#4r>a4.a#4.b4.R64
 55: p3v16b4.<d4.>a4.r64R2..
 56: |1p3v16r32b.:|
 57: |2p3v16<f#2.r4.>b2.r4.r<baagf#gdea2.r64r32p3v16f#1&f#2>R16
 58:
 59: (t4) o4v15q818
 60: @119<d2.r4.>g2.r4.rp2v14r64<baagf#gdea2.>R4.
 61: |:<f#2.R4.>b2.R4.<e2.&e4Rd2.R4.f#2.R4.>b2.R4.<e2.&e4R>a2.
 62: R64p3v16r16<c#&c#32>p3v16<c#2.e4.d4.>f#2.<g#4.e4.g#4.a4.
 63: >a2.r64p3v16<f#2.R4.>b2.R4.<e2.&e4R8d4.e4.d4.>R64
 64: p3v16<d4.f#4.c#4.R64>R2..
 65: |1p3v16r16<c#&c#32>:|
 66: |2p3v16<d2.r4.>g2.r4.rp2v14r64<baagf#gdea2.p3v16d1&d2>R16.
 67:
 68: (t5) o4v15q818
 69: @119p1v15r64>>da<df#a<df#4.>>gb<egb<eg4.>a2.>a4.df#a<df#a<d4.
 70: |:r64>>d8a<d<dc#d>a4.>gb<e<ed#ep3g4.p1>>a<c#ea<c#e>p3a4.
 71: p1>da<df#4.>a4.da<d<dc#d>a4.>gb<e<ed#ep3g4.p1>>a<c#ea<
 72: c#e>p3a4.p1d>af#d4.<<p3v16r16r32<dp3v16e2.g4.f#4.>a2.
 73: <b4.g#4.b4.R4.a2.R64>r64>>da<d<dc#d>a4.>gb<e<ed#eg4.>>
 74: a<c#ea<c#e>a4.>da<dc#ef#>b<df#<p3v16<g4.a4.R4.R64>R2..
 75: |1p3v16r16r32<d>:|
 76: |2p1v15r64>>da<df#a<df#4.>>gb<egb<eg4.>a2.>a4.<<d>af#
 77: d>af#d1&d2<<R16.
 78:
 79: (t6) o4v15q814.
 80: @119>>d<f#<f#>>g<g<g>a2.>ad<d<dr64
 81: |:>>d<f#a>g<gb>a<aa>d<f#>ad<f#a>g<gb>a<aad>dr64<<
 82: p3v16r8<e16.>p1v15r64>>18a8<c#eagec#4.>da<dbaf#d4.>eb<e
 83: <dc#d>b4.>a<c#eaab<c#del4.>>d<f#a>g<gb>a<aa>d<c#>b<<R64
 84: p1v15r64>>ga>a<<d8>a8f#8d2<<
 85: |1p3v16r8<e16.>:|
 86: |2>>p3d<f#<f#>>g<g<g>a2.>a<<d>d>d1&d2<<R16..
 87:
 88: (t7) o4v15q814.
 89: @119r8>>a<a2.>b<b2.r2.r>f#<f#r4r64<
 90: |:r8>>a<<c#r>>b<<d#r>c#<c#r>>ar2.a<<c#r>>b<<d#r>c#<c#r
 91: >>ar4r64<<p3v16r8r32<f#16>p3v16>>a4<e4g4c#>d4<d4a4d>e4
 92: <e4<c#4>b>a4<e4a8b8<c#8d8e8r64r8>>a<<c#r>>b<<d#r>c#<
 93: c#r>>a<ed4<R64p3v16>ga>a<d4>f#4rr64<<
 94: |1p3v16r8r32<f#16>:|
 95: |2r8>>a<a2.>b<b2.r2.ra>a2<<R1
 96:
 97: (t8) o4v15q814.
 98: @119r4>d<d2.>e<e2.r2.r>>a<ar8r64<
 99: |:r4>d<dr>e<er>e<er>dr2.d<dr>e<er>e<er>>f#r8r64<<
100: p3v16r8r16<g32>p3v16r8>c#4a4e2r8>a4<b4f#2r8>b4<<d4d2r8
101: >c#4a4g#8g8f#8e8r64<r4>d<dr>e<er>e<er>df#f#8<R64
102: p3v16>>ga>ar8<a4d2r64<<
103: |1p3v16r8r16<g32>:|
104: |2r4>d<d2.>e<e2.r2.r>f#>f#2<<R1R8...
105: (p)
```

リスト9　夢路より用カウンタ表示

```
1:000025BC 00000000  2:000025B6 00000000  3:000025B6 00000000  4:000025B6 00000000
5:000025B6 00000000  6:000025B6 00000000  7:00002571 00000000  8:000025B6 00000000
```

リスト10　NEW RALLY-X

```
10 '+----------------------------------------+
20 '|                                        |
30 '|          "Pop'n" NEW RALLY-X           |
40 '|                                        |
50 '|              (C) namco                 |
60 '|                                        |
70 '|          ARRANGED BY MIHO Y.           |
80 '|                                        |
90 '+----------------------------------------+
100 DEFSTR A-Z:DEFINT I,J,N,V:CLEAR&HFF00:CLS0:TEMPO0
110 "voice"
120 PLAY"T150 I1 P3 V14 O2 Q8";:A0="r1r1r1r1":'   BASS
130 A1="~6 L16 a8ega>cdeagar>c8<ag<a8ega>cdeagd#ec#8r8<a8ega>cde
agar>c8<ag<a8ega>cdeagd#ec#8r8_< L8"
140 A2="aa>a.a16r16<aa16>aa<aa>a.a16r16<aa16>aa<"
150 A3="aa>a.a16r16<aa16>aac#c#>c#.c#16r16<c#c#16>c#c#<"
160 A4="aa>a.a16r16<aa16>aa<ee>e.e16r16e>e16<g16>g16<g#16>g#16<<
"
170 A5="aa>a.a16r16<aa16>c#edd#e.e16g16>g16<g#16>g#16<a16>a16R<<
"
180 A6="dd>d.d16r16<dd16>dc#<<aa>a.a16r16ae16gg#<"
190 A7=">dd>d.d16r16<dd16>dd<<aa>a.a16r16<aa16>a16g#16a<"
200 A8="dd>d.d16r16<dd16>de<<aa>a.a16r16ae16g16>g16<g#16>g#16<<"
210 A9=">dd>d.d16r16<dd16>dc<<aa>a.a16r16ae16g16>g16<g#16>g#16<<
"
220 AA="aa>a.a16r16<aa16>aa\c#c#>c#.c#16r16<c#c#16>c#c#<"
230 PLAY A1;
240 PLAY A3+A6+A3+A8;
250 PLAY A2+A4+A2+A5;:PLAY A2+A4+A2+A5;
260 PLAY A3+A6+A3+A8;:PLAY A2+A7+A2+A9;
270 PLAY A2+A4+A2+A5;
280 PLAY A1;
290 PLAY A3+A6+A3+A8;:PLAY A2+A7+A2+A9;
300 PLAY A2+A4+A2+A5;
310 PLAY A1;
320 PLAY A3+A6+A3+A8;:PLAY A0+A2+A9;
330 PLAY A2+A4+A2+A5;
340 PLAY A1;
350 PLAY AA+A6+A3+A8;
360 PLAY ":I2 P3 V12 O3 Q7 K0";:'   MELODY
370 A1="L8 aaf#ear>c4<aaf#earf#4aaf#ea>cdd#e L16 d#ed#ed#ee4edc#
<b L8 aaf#ear>c4<aaf#earf#4aaf#ea>cdd#ae16d16c#<agg#ar"
380 A2=">e4c#ef#c#ef#c#<baf#a2f#aba>c#<bab>c#<a4f#e2"
390 A3="ageag#ededc<a>dc#<a4.>dc<aga>cdd#eg#edc#2"
400 A4="a{gag}8eag#eded{cdc}8<a>dc#<a4.>d{cdc}8<aga>cdd#eg#e{ded
}8c#2"
410 PLAY A0+A0+A0;
420 PLAY A1+A1;
430 PLAY A2+A2+A3+A3;
440 PLAY A1+A0;
450 PLAY A2+A2+A3+A3;
460 PLAY A1+A0;
470 PLAY A2+A2+"I3~3 O4"+A4+"I2_ O3"+A4;
480 PLAY A1+A0;
490 PLAY A2+A2;
500 PLAY ":I2 P1 V6 O3 Q7 K5 r8.";:'   ECHO1
510 PLAY A0+A0+A0;
520 PLAY A1+A1;
```

▶夢にまで見た「ファイナルファイト」が中古屋にあった。狂ったように喜んでいたら，値札に「14,900円」と書いてあった。即死状態である。TAKERUかなんかで安く再販してほしいっス。
福田　圭介(17)神奈川県

```
530 PLAY A2+A2+A3+A3;
540 PLAY A1+A0;
550 PLAY A2+A2+A3+A3;
560 PLAY A1+A0;
570 PLAY A2+A2+"I3¯2 O4 P3"+A4+"I2_ O3 P1"+A4;
580 PLAY A1+A0;
590 PLAY A2+A2;
600 PLAY ":I2 P2 V6 O3 Q7 K10 r8.";:'   ECHO2
610 PLAY A0+A0+A0;
620 PLAY A1+A1;
630 PLAY A2+A2+A3+A3;
640 PLAY A1+A0;
650 PLAY A2+A2+A3+A3;
660 PLAY A1+A0;
670 PLAY A2+A2+"I3¯2 O4 P3"+A4+"I2_ O3 P2"+A4;
680 PLAY A1+A0;
690 PLAY A2+A2;
700 PLAY":P1 K0"+A0;:'   BACKING#1
710 A1="I5 V12 O4 L16 Q8"
720 A2="I4 V8 O5 L16 Q8
730 A3="I6 V4 O3 L8 Q8"
740 A4="bbr4a8&a2aar4g#8&g#2f#f#r4f#8&f#2f#f#r4e8&e2"
750 A5="aaaa>aa8<a8aaa>aaaa<aaaa>aa8<a8aaa>aaaa<
760 A6="aaaa>aa8<a8aaa>aaaaeeee>ee8<e8ee8>edc#<b<"
770 A7="aaaa>aa8<aaaaaa>c#dd#>a8edc#8<a8g8g#8a8r8<"
780 A8="a1&a2&a>c4c#d1&d.d#16edeg4e16d16<"
790 PLAY A1+A4+A4;
800 PLAY A2+A5+A6+A5+A7;:PLAY A5+A6+A5+A7;
810 PLAY A1+A4+A4;:PLAY A3+A8+A8;
820 PLAY A2+A5+A6+A5+A7+A0;
830 PLAY A1+A4+A4;:PLAY A3+A8+A8;
840 PLAY A2+A5+A6+A5+A7+A0;
850 PLAY A1+A4+A4;:PLAY A3+A0+A8;
860 PLAY A2+A5+A6+A5+A7+A0;
870 PLAY A1+A4+A4;
880 PLAY":P2 K5"+A0;:'   BACKING#2
890 A4=">ddr4c#8&c#2c#c#r4<b8&b2bbr4a8&a2bbr4a8&a2"
900 PLAY A1+A4+A4;
910 PLAY A2+A5+A6+A5+A7;:PLAY A5+A6+A5+A7;
920 PLAY A1+A4+A4;:PLAY A3+A8+A8;
930 PLAY A2+A5+A6+A5+A7+A0;
940 PLAY A1+A4+A4;:PLAY A3+A8+A8;
950 PLAY A2+A5+A6+A5+A7+A0;
960 PLAY A1+A4+A4;:PLAY A3+A0+A8;
970 PLAY A2+A5+A6+A5+A7+A0;
980 PLAY A1+A4+A4;
990 PLAY ":I7 P3 V15 O2 L2 Q8";:'   SNARE
1000 A1="r4ccccccc¯3 P1 c16_ P3 c16 P2¯c8  P3_ "
1010 A2="r4ccccccc¯ P1 c16c16 P2 c16c16_ P3"
1020 A3="r4cccccc¯ P1 c8 P2 c8_ P3 c8c8c8r8 "
1030 A4="r1r1r1r.c16c16c16c16"
1040 PLAY A0+A1+A2+A1+A3+A1+A3;
1050 PLAY A1+A2+A1+A2+A1+A3+A0;
1060 PLAY A1+A2+A1+A2+A1+A3+A4;
1070 PLAY A1+A2+A4+A2+A1+A3+A0;
1080 PLAY A1+A2;
1090 PLAY ":I8 P3 V12 O2 L4 Q8";:'   BASSDRUM
1100 A1="crc8c8rcrc8c8rcrc8c8rcrc8c8r"
1110 PLAY A0+A1+A1+A1+A1+A1+A1;
1120 PLAY A1+A1+A1+A1+A1+A1+A0;
1130 PLAY A1+A1+A1+A1+A1+A1+A0;
1140 PLAY A1+A1+A0+A1+A1+A1+A0;
1150 PLAY A1+A1;
1160 PLAY":Y7,28 V9 O4 L1 Q8 K0 S0,0,0,0 =1^2";:'   BACKING#3(PS
G)
1170 A1="e4.e8&e2e4.g#8&g#2d4.d8&d2d4.c#8&c#2"
1180 A2="rrrrrrrrrrrrrrrr"
1190 PLAY A0+A1+A1+A2;:PLAY A1+A1+A2+A0;
1200 PLAY A1+A1+A2+A0;:PLAY A1+A1+A2;
1210 PLAY":V7 L1 O4 Q8 K3 S0,0,0,0 =1^2";
1220 PLAY A0+A1+A1+A2;:PLAY A1+A1+A2+A0;
1230 PLAY A1+A1+A2+A0;:PLAY A1+A1+A2+A0;
1240 PLAY A1+A1;
1250 PLAY":V10 L16 Q8 =3^1";:'   HI-HAT(PSG)
1260 A1="¯2c8_ccccrcrcrccrcrcrccccrcrcrccrcr"
1270 A2="crccccrcrcrccrcrcrccccrcrcrccrcr"
1280 A3="crccccrcrcrccrcrcrccccrcrcrcccccr"
1290 A4="crccccrcrcrccrcrcrccccrcrcrcrrr"
1300 A5="crccccrcrcrccrcrcrccccrcrcrcccccc"
1310 A6="r1r1r1r2.¯cccc_"
1320 PLAY A0+A1+A3+A2+A3;
1330 PLAY A2+A3+A2+A4;:PLAY A2+A3+A2+A4;
1340 PLAY A2+A3+A2+A5;:PLAY A2+A3+A2+A5;
1350 PLAY A2+A3+A2+A4+A0;
1360 PLAY A1+A3+A2+A5;:PLAY A2+A3+A2+A5;
1370 PLAY A2+A3+A2+A4+A6;
1380 PLAY A1+A3+A2+A5;:PLAY A6+A2+A5;
1390 PLAY A2+A3+A2+A4+A0;
1400 PLAY A1+A3+A2+A5
1410 END
1420 LABEL "voice"
1430 MEM$(&HB190,36)=HEXCHR$("F8 00 0A 70 30 00 24 2D 12 00 1F 1
F 5F 5F 12 0E 0A 0A 00 04 04 03 26 26 26 26 00 00 00 00 00 00 80
 00 00 00")
1440 MEM$(&HB1B4,36)=HEXCHR$("F1 00 0C 02 08 02 12 1E 32 01 1F 1
F 1F 1F 0A 05 05 05 00 01 01 01 88 28 28 28 00 00 00 00 00 00 80
 00 00 00")
1450 MEM$(&HB1D8,36)=HEXCHR$("F8 00 03 04 02 01 1E 1C 1A 00 DA D
D DC DF 0F 05 05 0D 05 04 02 03 21 11 31 12 00 00 00 00 00 00 80
 00 00 00")
1460 MEM$(&HB1FC,36)=HEXCHR$("E9 00 04 02 02 01 1E 23 30 00 1F 1
F 1F 19 01 03 03 0D 02 05 05 09 30 30 30 A6 00 00 00 00 00 00 80
 00 00 00")
1470 MEM$(&HB220,36)=HEXCHR$("FA 00 01 07 01 01 19 28 28 00 0F 0
F 0F 19 0A 0A 0A 05 02 02 02 02 56 A6 56 18 00 00 00 00 00 00 80
 00 00 00")
1480 MEM$(&HB244,36)=HEXCHR$("C7 00 31 72 24 68 07 07 07 07 1F 1
F 1F 1F 00 00 00 00 00 00 00 00 27 27 27 27 00 00 00 00 00 00 80
 00 00 00")
1490 MEM$(&HB268,36)=HEXCHR$("FC 00 40 40 40 40 00 00 00 00 1F 5
F 5F 5F 09 0F 14 0A 08 0E 12 0A A9 95 B8 99 00 00 00 00 00 00 80
 00 00 00")
1500 MEM$(&HB28C,36)=HEXCHR$("FB 00 0A 0D 00 01 0D 13 0B 00 16 1
A 1A 5E 00 1A 16 0E 00 C0 40 00 0A FA FB F7 00 00 00 00 00 00 80
 00 00 00")
1510 RETURN
```

(進)の「ちょっといいですかぁ?」

★スターウォーズ

オーケストラ演奏のコピー作品とは珍しいですね。SC-55は，弦や管などオーケストラ演奏に適した楽器音が多く内蔵されていますから，音色に困ることはほとんどないでしょうが，パーシャル数(同時発音数のようなもの)が24音と少ないため，決して得意なジャンルとはいえないようです。

このデータはその点をよくクリアしています。ところどころ音が薄かったり，パートが欠けてるなと感じたりする部分もあるのですが，編成に関しては大きな問題はないと思います。

ただ，音採りの間違いとおぼしき部分があちこち耳に障ります。原曲との些細な違いというよりも，どちらかというと音楽的にちょっとおかしいと感じられるものがいくつかあるのです。

ベース音やメインのブラスコードを間違ってしまうと，曲の要であるだけにごまかしが効きません。採譜にかなり苦労されたようですが，ここはもう少し気合を入れてやってみてはいかがでしょうか。

打ち込みの技術的についても，もう少し凝ったものを期待します。たとえば頭のブラスパートですが，ただ，

q7c4r {cc} c4r {cc}

とするよりも，

@q28c＊82q3{@u80c@u110c} ＊14@q35@u125c＊83q3 {@u90c@u110c} ＊13@u120

などとすると，より自然になるでしょう(リストが長くなるからヤだっていうのはナシよ)。

また，その場合さらに，

@y1,$63,61　　　/ TVF&TVAアタック

を加えて，音色のアタックを少し速くしておいたほうがいいでしょう(ここは好みの問題でしょうが)。

私の大好きな曲なのでチェックがいつもより厳しくなってしまいましたが，この曲に限らずオーケストラの再現では，このような工夫が確実に生きてきます。よりすばらしい演奏めざして頑張ってください。

★明日への扉

はじめにお断りしておきます。Z-MUSICの一部のバージョンで.sc55_chorusにバグが存在するようです。このデータはそのバージョンで制作されたらしく，正常な新バージョンでこの曲を聴くと投稿されたリストのままでは不都合が出てしまいます。そのため掲載にあたって42行目のパラメータを変更させていただきました。ご了承ください。

曲はうまくまとめられていています。SC-55にディストーションギターのミュート音がないのが悔やまれますね。主旋律(ピアノソロ，ギターを含む)のベタ打ちっぽさをなくせばさらによくなるだろうと思います。

いつもいっていることですが，人間の演奏を真似る場合，ただ音符を並べただけではマズイのです。これはオリジナルの演奏を完璧にコピーしろというのではなく，ピアノはピアノらしく，ギターはギターらしく鳴ってほしいということです。

Oh!X BOOKS「Z-MUSICシステムver.2.0」付属の「バランス.ZMS」を使って楽器間の音量を調整するとFM音源のボリュームがかなり大きくなります。お好みのバランスで聴いてください。

投稿と一緒に「楽器を2台同時に使うにはどうするのか」という内容の質問がありました。これは，パートの受信チャンネルを変えたり，OFFすることで可能だと答えておきましょう。機会があればくわしく解説します。

◇　　◇　　◇　　◇

今月はスペースの都合でここまで。山田さんの曲は私がバカでX1の操作がわからず聴けなくて残念無念。この次は念願のX68000で作れるといいですね。

古い話で恐縮ですが，神奈川の塩瀬さん，熱いメッセージをどうもありがとう。いい音を出す秘訣なんてこちらから質問したいくらいですが，一応「少しでも変だなと感じるところは残さない」ことだけ注意しています。(進藤慶到)

(善)のゲームミュージックでバビンチョ

西川善司

●WORLD HEROES & WORLD HEROES2 /SNK ADK
VHS:PCVP-11354　4,800円(税込)
LD:PCLP-00493　4,800円(税込)
ポニーキャニオン　12/17発売

「精霊戦士マッドマン」で一躍有名になった「ワールドヒーローズ１＆２」の攻略ビデオが山下章の監修で完成。この手のビデオでは定番の，ゲームキャラクターの声優たちが必殺ワザと連続ワザを紹介するという構成。笑いの要素が盛り込まれた「ワー・ヒー」ならではの，声優たちの漫才調の駆け引きが面白い。

お勧め度　7

●餓狼伝説SPECIAL～超絶武闘会～/SNK
VHS:PCVP-11355　4,800円(税込)
LD:PCLP-00494　4,800円(税込)
ポニーキャニオン　12/17発売

一発逆転のド派手な超必殺ワザで格闘ゲームに新たなゲーム性を見出した「餓狼伝説２」の続編，その名も「餓狼伝説SPECIAL」。NEO･GEO版発売とタイアップして山下章監修の攻略ビデオが登場。一撃必殺の連続攻撃は，スローモーションでボタン，レバー操作の字幕つきで解説。これはグッドアイデアだ！　ナレーションはゲームキャラクターの声優たち。隠れキャラやゲストエネミーのリョウ・サカザキとの対戦，爆笑の無意味ワザなども収録。

お勧め度　8

●VIRTUA RACING & OUT RUNNERS
CD:TYCY-5365～66　3,800円(税込)
東芝EMIユーメックス　発売中

セガの1993年の２大ヒットドライブゲーム「VIRTUA RACING」「OutRunners」の２枚組アルバム。ディスク１はアレンジバージョンで，ディスク２はオリジナルサウンド。アレンジバージョンは後半の超前衛音楽の２曲以外は最高の出来。トラック１「POLYGONIC CONTINENT」とトラック2「STREAM」は久々のヒットの予感。もうこれを聴かずにセガは語れずって感じ。オリジナルサウンドの「VIRTUA RACING」はファンファーレ的なものが多くて落ち着いて聴けない。「OutRun」の名曲３曲のOutRunnersバージョンは賛否両論だろう。怪しいコードワークが'90年代的という見方もある。

お勧め度　7

●SILPHEED
CD:TYCY-5338　2,800円(税込)
東芝EMIユーメックス　発売中

1993年夏，大旋風を巻き起こしたシューティングゲーム・メガドライブ版「シルフィード」のサウンドトラック。メガＣＤ版の曲をさらに発展させたアレンジで収録。非常にテンポの早い，リズムの激しい曲が多く，情景描写の曲がメインなので，ゲームをプレイしたことがなければそのよさは伝わりにくい。ボーナストラックにはPC-8801版の元祖「シルフィード」のオープニングとエンディングの新アレンジを収録。

お勧め度　7

●悪魔城ドラキュラX
CD:KICA7622～23　3,600円(税込)
キングレコード　発売中

PCエンジン版の「悪魔城ドラキュラ」はシリーズ10作目で「X」のロゴが入っている。X68000版にこのロゴが欲しかったが，それはさておき，この２枚組アルバムはドラキュラ尽くし。「悪魔城ドラキュラX」とメガドライブ版「VAMPIRE KILLER」からオリジナルサウンド(未使用曲1曲含む)をすべてと，X68000版の４曲をアレンジバージョンで収録。PCエンジン版はCD-ROM版からの収録で音質･アレンジともに一級品。

お勧め度　8

●ナムコビデオゲームグラフィティ Vol.10
CD:VICL40097～98　3,900円(税込)
テインメント　12/16発売

２枚組。ディスク１が「AIR COMBAT」「スーパーワールドスタジアム」「ブラストオフ」「ZOMBIE CASTLE」「エメラルディア」のオリジナルサウンドとアレンジバージョン３曲で計44曲。ディスク２は「Numan Athletics」「エクスバニア」「熱闘！激闘!!クイズ島」「爆裂クイズ魔Ｑ大冒険」「倉庫番DX」「スーパーワールドコート」のオリジナルサウンド計73曲を収録。

「AIR COMBAT」はCMソングタイプの押さえ気味のスマートなフュージョン系。インパクトは弱いがどれも聴きやすい。「Numan Athletics」はディスコミュージック系でゲームには合うが平常心で聴くには少々困難。「ブラストオフ」は念願のCDへの収録でファンは大喜びか。アレンジバージョンは「コズモギャングズ・ザ・ビデオ」がお気に入り。ナムコ伝統のメロディアスゲームミュージックを久々に聴いた感じ。

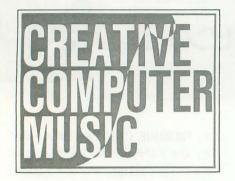
とにかく曲数が117と多い。これをお買得とみるかどうかは判断が難しいが……。

お勧め度　7

●ザ・コンピューターミュージック SC-55篇
CD:VICG5255　2,500円(税込)
ビクター音楽産業　発売中

サウンドクリエイター宗像仁志がローランドのGS音源モジュールSC-55のみを使用してラテン，ハウス，ファンク，バラード，アクロ，グランビート，クラシックの７ジャンルに挑戦。SC-55をプロが使ったらどうなるかを教えてくれるCDだ。音色名や音色の音域にこだわらない常識を超えた音の組み立ては既存のコンピュータミュージックの限界を打ち破った感じさえする。SC-55の音に飽きてしまった人は，これを聴いてSC-55の実力を再認識してほしい。

お勧め度　9

悪魔城ドラキュラX

Creative Computer Music入門(28)

偶成和音と和声進行の法則

先月号の予告を変更して，偶成和音の原理についてもう少し補足しましょう。原理編もそろそろ終わりに近づきつつあります。いままでの解説のなかでわからない部分が残っている人は，以前の号を読み返すなどして，ひとつずつ理解するようにしてみてください。

Taki Yasushi 瀧 康史

§Martika's kitchen

今月号を皆さんが手に取っている時期といえば，そろそろ，師走の忙しさが身に染みている頃でしょうか。クリスマス直前で，彼女や彼氏へのプレゼントを考えているのかもしれませんね。

さて，クリスマスに誰かに贈るのにふさわしそうなCDについて，ちょっとコメントしましょう。

そう，私はいま，1992年2月号の冒頭で，女性ヴォーカルの声でハリがあってよい，と紹介したマルティカの2枚目のアルバム「Martika's kitchen」を聴いているところです。

これは新譜というわけじゃないんですが，このアルバムが発売されたときは特に彼女の声を聴きたいという気分でもなかったので，アルバムを買わずにいたのです。私事ですが，実は私，再び引っ越しをいたしまして(連載中2度目になりますね)，新居での活力を得たいなと思ったところで，耳がマルティカの声を求めてきたってわけですね。

マルティカのCDを一度も聴いたことがなく，雑誌かテレビで名前を知ってるだけの人は，「マルティカ？ああ，そんなアーティストがいたねぇ……」というかもしれません。そうです。実は，前作のアルバム「魅惑のマルティカ」を発表してからすでに約2年ほどたっているのです。さらりといってしまえば，それだけですが，マルティカはこの「魅惑のマルティカ」がファーストアルバムで，このなかに収録されている「Toy Soldier」が実は全米ナンバーワンヒットになっているんですね。

そういう事実があるのならば，普通は2年もブランクをおかずに，みんながマルティカのよさを忘れていない時期，すなわち，最低でも1年以内に新譜を出すのが常識というものですよね。それがなぜ，第2作の発表までに2年もかかったのか……というのは，CDのジャケットを楽しみにしてもらうことにして，私は私の感じたことを書きましょう。

今回のCDを聴いて最初に思ったことは，色っぽくなった，ってことです。ハリのよさ，声のもつパワーは昔からの延長上にありますが，色っぽさというのは，少なくとも私は前回のアルバムでは感じなかったことです。女性ヴォーカルは色っぽくなくてはいけないというのが私の主義なんですが，これはかなり色っぽいですな。エッチな色っぽさを想像してしまうんじゃなくて，女性のもつ温かさとか，やさしさ，内面に秘めた力を感じさせる声に成長しています。

そう，成長。

ジャケットに書かれたコメント(日本版にしかないと思いますが)によれば，彼女はインタビューのときに，何度も自分が成長していることを強調していたそうですが，それは聴き手にもひしひしと感じられます。

クリスマスのムード作りに，なかなか適したCDだと思いますが，いかがですか？

§さらなる偶成和音

まず最初にあやまっておきましょう。大嘘つきになってしまいました。ごめんなさい。

前回の最後で，今月は「転調」について解説すると予告しましたが，ちょっと気が変わってしまいました。

先月号では，偶成和音についてかなり説明しましたが，残念ながらあれがすべてではありません。あらためて考えてみるに，この「原理編」は知識の穴を補充するためにやっているものですからね。やはり最後まできっちりやりましょう，ということで，偶成和音に関する原理をもう少し補充します。

その具体的な内容を挙げると，和音の転回形，変質和音，付加6，保続音，偶成和音が独立和音として認められるまで，そしてカデンツァの機能，といったところでしょうか。

転調については，来月きちんとやりますから，それまで楽しみに待っていてください。

それでは始めましょう。

§和音の転回

まず，「転回」について復習しましょう。

図1のような和音Cを例にとります。和音の構成音はCとEとGですね。この和音Cの基本形を転回したものが図の右側の2つで，第1転回形と第2転回形です。MML式表記をすると第1転回形が「EG<C」で，第2転回形が「G<CE」となります。そして，転回形というのは第2までしかありません。

これが基礎知識です。

1) 第1転回形

では，図2を見てください。

ここでは，単なるV→Iという進行のなかで，バスが経過的な変移をしています。

1つめの経過音は省略されたとします。

そこで，2つめの経過音に注目すると，動いたのはバスだけで，ソプラノ，アルト，テナーはVから動かずにそのまま残っていますから，これは結果的にはVの第1転回形になっています。この進行は基本的にはドミナントモーションなので，この第1転回形は，ドミナントモーションのなかでの「偶成和音」であるといえます。この偶成和音は，それ単体でもハーモニーとしておかしくはありません。もともと自然和音ですし(詳しくは，偶成和音が独立和音と認められるまで，の項を参照)。したがってこれは，独立和音であるといえます。

独立和音だということは，任意のときにいきなり登場させることができるわけですが，これは元来，いまいったような裏づけがあって構成されているということを忘れてはなりません。よって，この和音には特徴として，ドミナントモーションをしてしまうという法則性があることになります。

同様にこれは，VI^1→II，VII^1→IIIといったように，すべての和音においてドミナントモーションで進行することができます。ただし，IとIVの関係は，ドミナントモーションというよりも，トニックとサブドミナントという働きのほうが多いので，これは必ずしもドミナントモーションに限定されたものではありません。

2) 第2転回形

第2転回形にもこれと同様な法則が成り立ちます。

図3はドミナントモーションに第2転回形を加えた形ですが，この場合は省略されている経過音が2つになってしまうので，すでに経過とはいい難くなっています(実際に耳で聴いてみても，経過の省略とは思えないでしょう)。

したがって，これは第1転回形の場合に比べると，非常に劣った使い方といえます。つまりこれは，第2転回形があまり使用されないことを意味するわけです。

3) 付加和音の転回

付加和音というのは，4和音，5和音……のように，3和音に付加された形をもつ和音のことです。これについては先月号でも説明しましたね。付加された音は，和声音ではなく非和声音です。したがって，4和音が3和音＋経過音というように扱われるのと同様に，転回における法則も3和音の部分だけで扱われることを意味します。

図4-1 付加和音の転回

図1 和音Cの転回

図2 第1転回形は偶成和音である

図3 第2転回形

例を図4に示しておきましょう。

経過音1は省略され，経過音2は4和音を作るための経過音です。最初のVにおいて5度であるDが省略されているのは，根音が3つ重複しているからで，省略するのは，3度の音よりも5度の音のほうがよいからです。ソプラノがいきなり，B→Dと進行するのは，最初の和音で省略されたDを補充するためと，Bが根音で使われたためです。このような進行が，結果として，図4-2のような最も基本的なドミナントモーションになるわけです。

§変質和音

変質和音というのは，いままで言葉では触れていなかったものの，実はちゃんと扱っています。

変質和音が生まれた理由は，いままで述べた転回形と

図4-2 基本的なドミナントモーションになる

同じ理由で，ドミナントモーションをもっと円滑に行おう，といった要因からです。転回形と違う点は，転回形が全音階的な経過音を独立させて作成したのに対し，変質和音はこの経過音に半音階的な要素を加えてできているということです。

ドミナントモーションの基本が，Vの3音であるBすなわち導音が，Iの根音への進行を促すことだというのは，以前からいっていたとおりです。V_7，V_9においてもこのことは同様で，これらのおかしな和音が誕生したのも，ひとえに，それぞれ，Iの3音，5音への進行を促すためにあると断言できます。

Vの根音はバス進行の変移を表す音という使命があるため，これも非常に重要な役割をもちます。

しかし，Vの5音は依然として，Iへ進行するための仕掛けにはならないので，ドミナントモーションにおいては，省略されることもしばしばあります。これが先ほど説明した図4-1の最初のVですね。

では，このVの5音に「指向性」という強い意味をもたせるためにはどうすればよいでしょうか。それには，「経過」というのが進行の手助けになるように，ここでも同じように，半音進行をさせ，Vの5音(D)を次のモーションで半音高めたり，低めたりして，Iの根音や3音へうまくつなげるという方法があります。そうすればVの5音の存在価値はよりはっきりしてきます。

図5を見てください。

最も基本的なドミナントモーションですが，ソプラノのDはD♯を経てからEに上がっています。ここでの経過和音は$Gaug_7$になっています。

ちなみにこの進行を見てすぐに，このドミナントモーションは次への期待が予想できると思った人は，よく勉強している人ですね。

図5 半音進行で指向性をもたせる

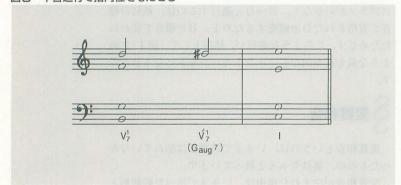



図6 付加6の生成

§付加6

付加6の生成も，要は変質和音と同じようなものです。

図6を見てください。これはサブドミナントからトニックへの進行ですが，途中の経過音はサブドミナントに乗ることにより6となります。

これについてはただの「変質」なので，あえて説明はいりませんね。

§保続音

付加6の和音を除いて，ほとんどすべての偶成和音はV→I進行，つまりドミナントモーションを装飾してできています。これはそれだけ，V→Iという進行が装飾しやすいことと，また，逆に調の根本となる進行であるため，装飾されても本質的な機能を失わないということを意味しています。

しかしながら，和音の本質はやはり根音です。したがって，いくら頭にいろいろな付加音がついたにしても，根音ははっきりと刻むのがベストといえます。

その行為をよりいっそう充実させるのが，根音保続というわけです。

今回の例のなかでは，図2，4-1，4-2，5において，Vの根音であるGがそのままIの5音に保続しています。

§偶成和音が独立和音と認められるまで

前回から2回にわたって，独立和音である3和音の上に，いくつかの非和声音を付加し，偶成和音を作り出すということを行ってきました。

このなかで，あるものは独立和音として扱われ，あるものは偶成和音のままとして扱われます。独立和音の定義はそれ1つだけで意味をもつことです。具体的にいうと，ドミナントモーション，サブドミナントモーションのなかで経過的に生成される音は独立和音になりがちです。そして，そうして生まれる独立和音は，一般に，それ単体で扱われても十分機能し，もともと経過のために必要であった，手前の$V_{(7)}$を必要としなくなり，やがて独立します。

§カデンツァの機能

さて。

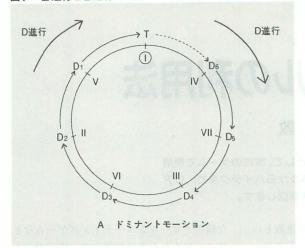
ここまで読んできた方にはもう，ある程度わかると思いますが，ハーモニーの進行には根底に流れるドミナントモーションを応用したいくつかの法則が成り立っています。

図7を見てください。

この図は，音楽を勉強するためにいろいろな本を読んでいる人ならば，どこかで必ず一度ぐらいは見たことがあると思います。

図7-Aはドミナントモーションの進行で，これを楽譜

図7　D進行とS進行

A　ドミナントモーション

B　サブドミナントモーション

に表すと，図8のようになります。

基本的にⅠは直接どこに進んでもかまいませんが，いったんドミナントモーションに入ると，あとは矢印の順に進むことが同調においての進行の原則となります。つまり，Ⅰ(T：トニック)からⅡ(D_2)に進むと，そのあとはⅡ(D_2)→Ⅴ(D_1)→Ⅰと進みます。Ⅵ(D_3)に進んだ場合は，そのあとはⅥ(D_3)→Ⅱ→Ⅴ→Ⅰとなります。したがって最後のⅠからはⅤ(D_1)がいちばん近く，Ⅳ(D_6)がいちばん遠くなります。

このなかで，Ⅶは減3和音をもち，不安定になります。したがって，これが独立的にD_5として扱われるのはいささか不適当で，これは特別な理由がない限り，単体では使用されません。使われるのはゼクエンツという強力な裏づけがある場合です。

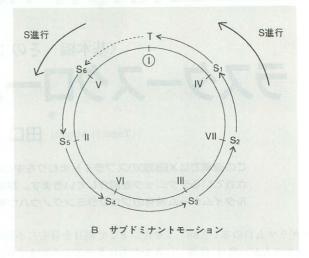
さて，Ⅶが用いられないので，Ⅶに進むⅣ(D_6)はD_6としては利用されにくくなります。したがってⅣはたいていの場合，サブドミナントとして扱われるか，Ⅱの代理としてD_2として利用されます。

計算でいくとD_7はⅠなのですが，Ⅰはそれだけで安定するので，D_6に流れる必要性はなく，これはD_7と解釈するには不適当です。

一般にⅤ→Ⅵの関係はD_1→D_3となりますが，根音省略形においてD_1はD_4と解釈できます。D_1は機能的に非常に強い性格をもち，根音を省略されてもされなくても，あるいは先がD_4であったとしても，Tとみなして進むことができます。これが，ⅥがⅠの代理として扱われる裏づけになります。

Ⅲは，Ⅶの解決和音ですが，Ⅶはあまり使われることがありません。さらにⅢはⅥへのDなのですが，上の理由から，　ⅥはTとして扱われることが少なくなるので，ⅢがD_4として考えられることはまずなくなってしまいます。

結果として，まっとうに用いられるドミナントモーションは，D_2とD_1ぐらいのものになってしまいます。

一方，図7-Bのサブドミナントモーションでは，S_6→S_5のところにドミナントモーションを裏切る進行があり，これは拒絶されます。S_2→S_1のあいだにも同様な理由で拒絶があります。さらに途中のS_5→S_4などの進行は，近親調である短調への借用とみなされてしまいます。というように，結果的にはサブドミナントモーションは理論と矛盾して，S_1→Tの進行しか利用できなくなってしまいます。

これを整理すると，7つの和音は，T，D_2，D_1，Sの4つの機能に還元でき，それ以外はその代理であるといい切ることができます。

つまり和声進行の最も基本となることは，

・T　T，D_2，D_1，Sのいずれにも進むことができる
・D_2　D_1のみに進行する
・D　Tのみに進行する
・S　DかTに進行できる

となり，これは連載の最初にお話ししたカデンツァの法則にたどり着くというわけなのです。

§おわりに

原理編もいよいよ大詰めになってきました。

来月こそは，転調について扱います。あと2回ぐらいで，和声学の原理をだいたい説明し終えることになりそうです。

これで，丸3年近く続いたこの連載もいよいよ幕を閉じることになります。いま思えば，読者の皆さんにとって何にも力になれなかったような気がします。初志貫徹できなかったかなぁ？　できなかったよなぁ……。

それでも，1度も休んでないよな。たしか。うん。休んでない(あと2回で休んでしまったらアウチか(笑))。

皆さんも，あともう少し頑張ってください。

図8　D進行

"実戦！"ゲーム作りの

KNOW HOW

〈基本編　その1〉

ラスタースクロールの利用法

Taguchi Atsushi 田口　敦

この連載ではX68000のスプライトまわりを中心として，実際のゲームで使用されているテクニックを紹介していきます。基本技からハイテクまで，リアルタイムゲーム特有のプログラミングノウハウを解説します。

ゲーム好き&プログラマ人口の多いX68000のユーザーなら大半の人が一度は「世界一面白いゲームを作ってやるぜぇ」などと理想に燃え，なんらかのコンピュータ言語に触れたことがあるのではないかと思います。

しかし，BASICやCでは速度的な問題があったり，相当な上級者でもない限り標準外部関数/ライブラリの機能内のことしかできないということもあり，ちょっと本格的なゲームプログラム（特にアクションやシューティング）を作成すると，すぐに限界が見えてきます。そして，その壁を見てしまった人のほとんどはアセンブラへ移行しているのでしょう。

実際にアセンブラを使ってプログラムを組み始めると，確かに自由度はほかの言語と比較にならないくらいあるのですが，それだけに持て余してしまう人も多いのではないかと思われます。

特に初級・中級の人は「××ゲームの○○みたいなことをやってみたい」と思ってもどうしたらいいかわからず，結局，C言語をコンパイルしたようなプログラムを書き，「なんだこれならCで組んでも変わらないじゃないか」とアセンブラに冷めてC言語に戻ってしまう人もいるのではないでしょうか。

そんな初級・中級者の人たちにゲームを作るうえでのいろいろなテクニックを提供したいと思います。タイトルの“実戦！”とは実際にプログラムと戦う人たちへ。“KNOW HOW”とは（読めない人はあまりいないと思いますが「ノウハウ」と読みます）日本語に直訳すると技術的提供という意味になります。

実を申しますと，私も現在プログラムと戦う人であり，勉強中の身なので（アセンブラ歴1年と半分）ほかのライターさんのように高速開発爆速執筆という技ができません。そこで隔月を目安に不定期連載というかたちにさせてもらいますが，その分，できるだけ密度の高いものを書かせてもらいたいと思います。

まあ前置きはこのぐらいにしておきまして，この企画の意図するところを説明しましょう。読者の皆さん，特にゲームプログラマとして初級・中級の人はゲームを作っている最中に「市販ゲームで使われている効果と同じことをしたいけど，やり方がわからない」なんてことがいくらでもあると思います。

私だっていまでもそういうときがあります。そんなときはたいてい雑誌の技術特集みたいなものを探してそれを参考に問題を解決したりするのですが，ほしい資料がまとめて載っていないことも多く，探すだけですごく手間がかかったりします。

そこで，比較的難易度が高くて頻繁に用いられるような技術を小出しにしながら提供しようと思ったわけです。先ほどいったように私は完成されたプログラマではなく，世の中に遅れない程度に勉強しているただの学生です。私が提供するのは見本ではなくて，レベルを上げるための踏み台のようなものです。皆さんのレベルを上げるためならよろこんで踏まれましょう。

とりあえず実際の内容ですが，10回くらいに分けてアクションやシューティングゲームなどによく使われるテクニックを多少我流を含めまして提供したいと思っています。

◆ラスタースクロールとは

さっそくですが，基本編としてラスタースクロールをやらせてもらいたいと思います。

さて，ラスタースクロールを知らない人はいるでしょうか。ダライアス2に始まった横スクロールシューティングゲームなどによく見られるあの背景のゆらゆらのことといえばわかると思います（わからなかったらごめんなさい）。実は，ラスタースクロールはほかにもいろいろ利用価値があります。ストIIの床とか，レーシングゲームの道（ポリゴンものを除く。ちなみにアウトランナーズとかはすごいですね）など，流行のゲームにはよく使われるものなんですね。この記事を読み終わる頃にはあなたもラスタースクロールマスターです（だといいな）。

基本編なのになぜいきなりラスタースクロールから入るのか疑問に思う方もいるのではないでしょうか。昔から（といってもここ10年もたってないが）ラスタースクロールというものは相当なテクニシャンでなくてはできないなどといわれてきました。でもこれはラスタースクロールが生まれたばかりの頃の話。そう，ラスタースクロールというのは基本ができていれば簡単なものであって，逆にいえばラスタースクロールを覚えることによって割り込みやアドレッシングモードなどといった基本をマスターすることができるのです。

それではいったいなにが難しいのかというと，ラスタースクロールの発想がコロンブスの卵的ですごいのです。現れた当初，まさかラスターごとにスクロールさせているなどと思いもよりませんでした。また，そういう発想があったとしても，当時の“マイコン”にはそれだけの機能も処理速度もありませんでした。

しかし，いまの子供たち（大人も含めて）は恵まれています。その発想やアルゴリズムなどはこの本の記事として提供されますし，あなたがOh!Xを読んでいるとその横にはそれを実現できる夢の機械があります。あとはやる気さえあれば目の前の壁は崩れ去るのです。あなたも夢への道をズンドコ

突っ走りませんか。

◆ 基本のお勉強

ラスタースクロールをマスターするためには，まずCRT（ブラウン管）の構造を知らなければなりません。これまでにも何度か解説した記事がありますが，軽くおさらいしておきましょう。

CRTの中には電子銃があります。皆さんが想像するとおり電子を発射する装置です。そこから光の30%の速度で発射された電子がCRT表面のガラスに塗ってある蛍光塗料にぶつかると光を発します。このとき電子の量が多いと明るく，少ないと暗く映ります。

電子銃から発射される電子は1本のビーム状であり，画面の左上から右に電子銃が走査することにより1ラスターの表現ができます。

この動作が画面の上から下まで行われることにより1枚の映像ができあがります。一見ちらついてしまいそうな構造ですが，十分に高速だと残像現象で人間の脳が1枚の画像だと勘違いしてしまうので，こういった機構が成立しているのです。

電子銃が画面の左から右に電子を照射している時間を「水平表示期間」といい，電子の照射をやめて右から左へ帰る時間を「水平帰線期間」といいます。同じようにラスターが上から下へ表示される期間が「垂直表示期間」で下から上へ帰る期間を「垂直帰線期間」といいます（図1）。

ラスタースクロールとはこの水平帰線中に各画面のスクロール値（表示先頭位置）を変えることによって実現できるのです。X68000の場合，最大でグラフィック4プレーン，テキスト画面，BG2プレーンが独立にスクロール指定できます。ただし，グラフィックやテキスト画面では縦方向のラスタースクロールができませんので，以下ではスプライトBGを使用したラスタースクロールを中心に解説します。

さて，ここまでは比較的理解が容易だろうと思われますが，1993年10月号のFISH.Xの記事で横内氏が話題にしているように，割り込みの理解と扱い方が難しいのではないでしょうか。これもおさらいする必要がありそうなのでさらっと書きます。

まず割り込みっていったいなに？　というところから入らねばなりません。特に初心者はここで引っ掛かると思います。

“サブルーチン”という言葉は知っていますよね。割り込み処理というのはあるきっかけによりサブルーチンを自動実行する仕組みのことです。そのきっかけというのがキーを押すことや一定時間ごとのタイマであったり，有無をいわさぬリセットだったりします。

通常のプログラムの処理中にこれらのきっかけがあると，設定された割り込み処理ルーチンが立ち上がり，処理が終了するとなにくわぬ顔で処理を元のプログラムに戻します。

つまり，割り込み処理を一度設定してしまえば，プログラム上で意図的にルーチンをコールしなくても，きっかけが起こるごとに自動的にルーチンを実行してくれます。なんて便利なんでしょう。もし，この割り込みというものがなかった場合，そう，たとえばタイマ割り込みがなかったら「一定時間たったかどうか判定して時間がきたらルーチンを実行する」なんてことを，メインループやあちこちのサブルーチンの中で組んでやらなければなりません。

これでは時間待ちの部分だけでCPUパワーを食ってしまい，並行して「音楽を鳴らす」とか「カーソルを点滅させる」とかいう処理ができません。割り込みがあるおかげでまったく別の処理を行っていても，プログラマの関知しないところで条件さえ満たされれば勝手にルーチンを実行してくれます。

ラスタースクロールを行うために必要なきっかけは，「CRTのラスター走査が始まったよ」という信号と「ラスター走査を画面の上から下まで行ったので電子の照射をやめて上に戻るよ」という信号です。

これらは水平帰線期間と垂直帰線期間と呼ばれているのは前に書きましたが，これを割り込みのきっかけとして使用します。ゲームを作るうえでは当たり前の機能ですので，X68000にはこれらの割り込みを利用する機構がありますが，ほかのパソコンではこういった機能をまったく持っていないものもあります（特に水平帰線割り込みはほとんどの機種が持っていない）。

割り込みの設定の仕方は通常はIOCSコールを利用します。IOCSを使用しないと例外ベクタやインタラプトイネーブルレジスタなど，CPUやMFPまで勉強しなくてはならないのでここは自由研究とします（『InsideX68000』を見てください）。

さて，IOCSコールの載っている本をペラペラめくっていると，なんかそれらしきものが見つかりました。VDISPST，CRTCRAS，HSYNCSTの3つです。

まずVDISPSTの説明を見ると垂直周期による割り込みと書いてあります。このコールはデフォルトで垂直帰線期間で割り込み動作を行うようになっているので特に問題はないでしょう。問題なのはほかの2つのコールです。

マニュアルを見るとCRTCRASの説明のところに「ラスター走査による割り込み」と書いてあります。これは一般にラスター割り込みと呼ばれるもので，なんかラスタースクロールに関係ありそうですが，これは「指定した回数ラスター動作を行ったら割り込み動作を行え」というもので，上下分割スクロールなどには向いていますが，ラスターごとにスクロール値を変えるラスタースクロールにはあまり向いていません（できないことはない）。

そこでHSYNCSTを使うわけですが，これはラスター走査が行われるごとに割り込みを発生させるものです。俗に水平同期割り込みと呼ばれます。

◆ アルゴリズム

それでは割り込みの説明が終わったところで早速，実際の動作に入りましょう。

ざっとプログラムの流れを書くと，

●メインプログラム

1)　ラスターごとのスクロールデータを用意

2)　垂直帰線の割り込み処理を設定

3)　水平同期の割り込み処理を設定

4)　通常処理（キーデータ取りやスプライトの移動計算など）

5)　垂直帰線待ち

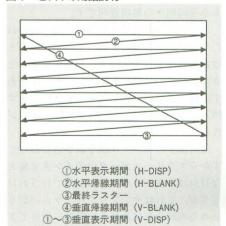
6)　4)に戻る

●水平同期割り込みの処理

1)　ラスター番号に当たるスクロールデータをスクロールレジスタに入れる

2)　スクロールデータのポインタを更新す

図1　CRTの用語説明

る

●垂直帰線割り込みの処理

1) ラスタースクロールデータのベースポインタ更新
2) メインプログラムに垂直帰線期間がきたことを教える

となります。

では，これらを順を追って説明しましょう。まずはメイン部分から。

1)のラスタースクロールのデータ（スクロール位置）を用意するというのはラスターごとに変化させたい以上，ラスター本数だけ用意しなければなりません（図2）。

ラスターの本数は31kHzで568本，15kHzで260本です。また，ゆらゆらラスタースクロールのように縦方向にスクロール値を変化させる場合は，同じデータを2つ続けて持ち，スクロールデータのベースポインタをずらしていくことにより実現します。

2)と3)は，先ほどいったようにIOCSに割り込みルーチンのあるアドレスを渡せばOKです。あっ，そうそう垂直帰線割り込みは毎回行うようにカウンタに1を指定しておいてください。

4)の処理が重要です。ここでは通常割り込みが処理に使用するデータなどを作ります。特にラスターの具合（ゆらゆらの振幅など）を変える場合次の垂直帰線割り込みがくるまでに処理を終えねばなりません。しかし，たいていの場合は垂直帰線がくるまでいくらか時間が余るので垂直帰線待ちをします。それが5)です。

ここまできたらまた4)に戻ります。え？これでは永久ループではないかと思った人は4)の処理中に終了条件が満たされたらプログラムを終わりにするという処理を付け加えてください。このとき割り込みの状態を初期化するのを忘れないように。

＊　＊　＊

さて，ラスタースクロールの本体にあたる水平同期中の処理説明です。

1)のラスター番号というのは現在行われたラスター走査回数と同じものです。あとはわかりますね。スクロール値を変えるときはIOCSなど無駄の多いルーチンは使わず直接スクロールレジスタに書き込みます。BGおよびグラフィックのスクロールレジスタアドレスは図3,4を見てください。

毎回違うスクロール値を書き込むわけですからスクロールデータのポインタを変えなければなりません。それを2)で行うわけですが，これはサンプルプログラムの割り込み処理部分を見たほうが簡単に理解できるでしょう。

図2

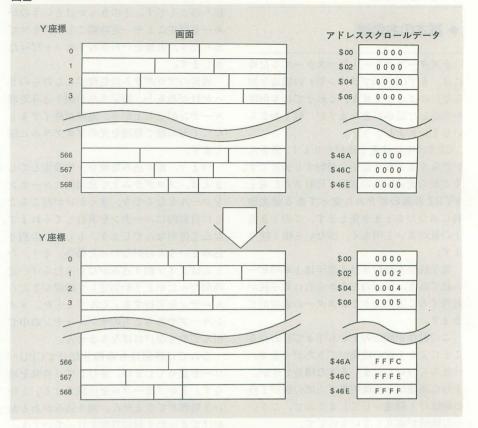



図3　グラフィックスクロールレジスタ（$E80018～$E80026）

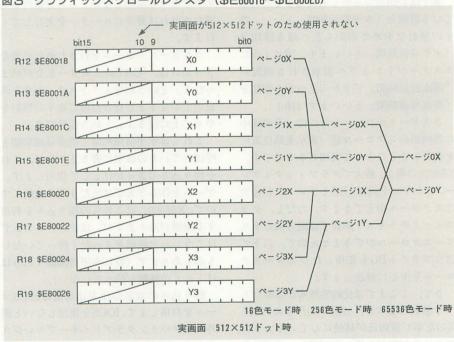
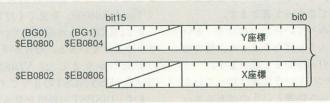



図4　BGスクロールレジスタ

最後に垂直帰線割り込みでの処理です。垂直帰線処理中のベースポインタ更新というのは水平同期処理中にスクロールデータのポインタを変化させてしまうので元に戻して次の表示期間に備えるというだけのことです。

次に重要なのが，メインプログラムに垂直帰線期間処理が行われたことを教えることです。これは共通のワークになにか書き込む程度の処理でよいのですが，これがないと表示期間中にメインプログラムが何回もループしてしまうことになります。プログラムをうまく動かすためには垂直表示に同期させねばなりませんから，忘れないようにしてください。

もうひとつ忘れちゃならないのがメインプログラム側で先ほどのワークをクリアしておくことです。私はいまでもこれをよく忘れます。おかげで原因不明のバグに１日中苦しめられて「うきょぉ！」状態になったりしますが，皆さんはこんな馬鹿な真似はしないでくださいね。

◆ サンプルプログラム

やはりこの手のものは実際のプログラムの動きを追ってみないことには完全な理解は不可能です。私のソースはまだまだ未熟ですが，できるだけわかりやすく応用がきくように作成したつもりです。

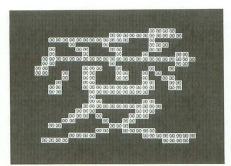
まずリスト１ですが，これはBG表示のためのサブルーチンです。特に説明はいらないと思いますが，気にいらない人は自作してください(責任転嫁)。アセンブルしたあと，できあがったオブジェクトファイルをリスト２またはリスト３とリンクしてください。

リスト２は波型ラスタースクロールです。一見お馴染みのもののようですが，よく見ると縦ラスタースクロール（以下縦ラスター）もしています。催眠術にも使えそうな気がするほど気味が悪いですが，片思いの女の子を洗脳するのに使えそうです。

この縦ラスターというのがくせものでグラフィック画面のスクロールレジスタはラスターごとの縦スクロールができません。BGのみの芸当です。私は昔これでハマりました。

プログラムの説明ですが，波の振幅をカーソルで変えられる以外は非常にオーソドックスな作りだと思います。波の振幅を変えるときのデータ減算は固定小数点演算をしているのでわかりにくいと思いますが，これを参考にマスターしてしまうのもひと

これが元の形

波状にうねらせる

つの手です（少々変なことをしているので参考にならないかもしれない)。

自分のプログラムに応用したい人は，波の動きや振幅をリアルタイムで変えられる部分を取り去ってしまえばそのまま使えると思います。このプログラムのメイン処理は，新しい波データを作ることとESCキーが押されたら終了処理ということしかやっていませんので処理時間はたっぷり余っています。ここにいろいろな処理をつけ加えると垂直同期動作の勉強ができます。

データ領域にあるサインテーブルですが，このプログラムではテーブルの半分しか使用していません。用途がいろいろあるのでほかのプログラムでも使用してみてください。

さて，もうひとつのプログラム（リスト３)ですが，これはストIIの地面と同等の処理です。

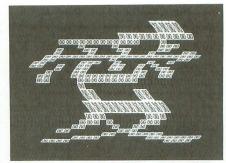
場合によってはこっちのプログラムのほうがわかりやすいかもしれません。

細かい説明をすると，初めにスクロールさせるためのデータを一気に作ってしまい，左右キーが押されることによってそのデータを減算しています。

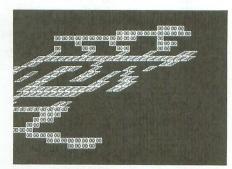
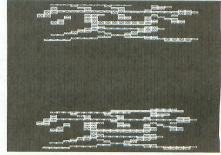
ここでも固定小数点演算を使用しています。上位16ビットが整数部，下位16ビットが小数部となっています。もし，整数演算しかしなかった場合どうなるか，想像がつきますでしょうか。

２つのプログラムともデータレジスタをワークとして使用していますが，これは高速化のためです。水平同期期間というのは非常に短いので，いちいちメモリからアクセスしていては処理に無駄が出てしまいます。

しかし，データレジスタはメインプログラムのほうで使用しているかもしれないので，割り込み内でワークとして使用するときはほかのプログラムでそこのレジスタを使用していないことを確かめてください。

さらに，反則に近い技ですが，ちらつきを抑えるためにタイマC，Dの割り込みを止めています。これを止めないと，どのようにちらつくのかは各自実験してみてください。

おっと，いい忘れるところでした。リスト３はちょこっと変更するだけでまったく別の効果が表れます。29・31・119行のアスタリスクのついている行をその直前の行と交換してみてください。交換後アセンブルして実行してみるとなんと，縦方向拡大縮小プログラムになります。

ハッキリいってモドキですが，これをうまく使うことにより拡大縮小はもちろん，レーシングゲームの地面の起伏などにも使えます。アルゴリズムは簡単で，ラスターごとに１ドット以上縦スクロールさせると縮小。１ドット未満の縦スクロールだと拡大になります。あとはこれも自由課題。実際に触れてみることがいちばんだからです（ということにして逃げよう)。

もうひとつ。ZP.Xのような垂直同期割り込みを使用するアプリケーションが常駐しているとIOCSで割り込みの設定ができま

疑似的な遠近感をつけてスクロール

縦に拡大する

せん。おそらくZP.Xなどが割り込みを乗っ取ってしまうためと考えられます。

もし，このプログラムを実行したとき，なにも表示されずに戻ったら常駐物のせいで割り込み設定ができない可能性があるので確認してみてください。

◆ 今後の予定

最後に今後の予定です。

いきなり初回にラスタースクロールがきたのでびっくりした人もいるのではないでしょうか。次回はたぶんもう少し簡単な内容になると思います。

その内容というのはBGのマップについてです。つまり，BGの実画面より大きな画面を使うときどうしたらよいかというものです。結構利用価値があると思いますが，どうでしょうか。

そのあとは巨大スプライトの使用方法(PCG転送によるアニメーションやキャラクター管理）など市販ゲームなどに実際に使われるようなテクニックをいくつか書かせてもらいたいと思います。

一般的なテクニックをひととおり提供したあとは，一般的でないテクニックも考えています。ちょこっと公開してしまいますと，スプライト表示数の倍化やスプライトの拡縮回転などです。まだ完成の域には達していませんが，これらは皆さんと一緒に勉強しながら進めたいと思います。

そして最終的には究極のゲームマネージメントシステムを組み上げることができたらいいなぁと思っています。実は，現在リアルタイムゲームの開発効率を大幅に上げることのできるオブジェクト指向(？)タスクコントロールシステムなるものが動いているのですが，まだまだ完成はほど遠く，連載を進めながら制作していきたいと思っています。

悪口でもなんでも読者の声はどんどん参考にさせてもらおうと思っているのでお便りをいただけたらうれしいですね。それではまた。

協力　H．C．S．とその仲間たち

リスト1

```
  1: *
  2: *        愛を表示するためのプログラム
  3: *
  4: *
  5:
  6:          .include        iocscall.mac
  7:          .include        doscall.mac
  8:
  9:          .xdef           screen_init
 10:
 11: BG0ADR          equ     $EBC000         * BG_VRAMアドレス
 12: G_VRAM          equ     $C00000         * G_VRAMアドレス
 13: BGPALET         equ     $E82200+$20     * パレット1
 14: BG_CTRL         equ     $EB0808         * BGコントローラー
 15:
 16: SP_BG_ON        macro
 17:         ori.w   #%0000_0010_0000_0000,BG_CTRL
 18:         .endm
 19:
 20:         .text
 21:         .even
 22:
 23: screen_init:
 24:         IOCS    _SP_INIT
 25:         IOCS    _SP_ON
 26:
 27:         moveq.l #0,d1
 28:         moveq.l #0,d2
 29:         moveq.l #1,d1
 30:         IOCS    _BGCTRLST
 31:
 32:         andi.w  #%1111110111111111,$eb0808      *BGOFF
 33:
 34:         clr.w   d1
 35:         move.w  #256,d2
 36:         IOCS    _BGTEXTCL
 37:
 38: fontget:                        * FONTデータをBG_VRAMに吸いだす
 39:         lea.l   aisym,a1
 40:         IOCS    _SYMBOL                 * ビットデータをワードデータに変換
 41: fontput:
 42:         clr.w   d3
 43:         lea.l   BG0ADR+(64*4+4)*2,a1
 44:         lea.l   G_VRAM,a2
 45:         move.w  #24-1,d0
 46: putloop00:
 47:         move.w  #24-1,d1
 48: putloop01:
 49:         move.w  (a2)+,d3
 50:         add.w   #256*1,d3
 51:         move.w  d3,(a1)+
 52:         dbra    d1,putloop01
 53:         add.w   #(64-24)*2,a1
 54:         add.l   #(512-24)*2,a2
 55:         dbra    d0,putloop00
 56:
 57: pcgset:                         * PCG生成
 58:         move.l  #0,d1
 59:         move.l  #1,d2
 60:         lea.l   pcgdata1,a1
 61:         IOCS    _SP_DEFCG
 62:
 63:         move.l  #1,d1
 64:         move.l  #1,d2
 65:         lea.l   pcgdata2,a1
 66:         IOCS    _SP_DEFCG
 67:
 68: paletset:                       * カラーデータ生成
 69:         move.l  coldata,BGPALET+2
 70:         move.l  coldata+4,BGPALET+6
 71:
 72:         ori.w   #%0000001000000001,$eb0808      *BGON
 73:
 74: aiquit:
 75:         rts
 76:
 77:         .data
 78:         .even
 79: pcgdata1:
 80:         .dc.l   $44333333,$44333333,$44333333,$44333333
 81:         .dc.l   $33443344,$44334433,$44334433,$44334433
 82:         .dc.l   $44333333,$44333333,$44333333,$44333333
 83:         .dc.l   $44333333,$44333333,$44333333,$44333333
 84:         .dc.l   $44333333,$44333333,$44333333,$44333333
 85:         .dc.l   $33443344,$44334433,$44334433,$44334433
 86:         .dc.l   $44333333,$44333333,$44333333,$44333333
 87:         .dc.l   $44333333,$44333333,$44333333,$44333333
 88: pcgdata2:
 89:         .dc.l   $12222222,$21111112,$22111111,$21111121
 90:         .dc.l   $21211111,$21111211,$21121111,$21112111
 91:         .dc.l   $21112111,$21121111,$21111211,$21211111
 92:         .dc.l   $21111121,$22111111,$21111112,$12222222
 93:         .dc.l   $22222221,$21111112,$11111122,$12111112
 94:         .dc.l   $11111212,$11211112,$11112112,$11121112
 95:         .dc.l   $11121112,$11112112,$11211112,$11111212
 96:         .dc.l   $12111112,$11111122,$21111112,$22222221
 97:
 98: coldata:        *- G -* R */ B /I
 99:         .dc.w   %0000000011111110
100:         .dc.w   %1111111111000000
101:         .dc.w   %0000010011000100
102:         .dc.w   %0000000111000000
103:
104: aisym:
105:         .dc.w   0,0
106:         .dc.l   aifont
107:         .dc.b   0,0
108:         .dc.w   0001
109:         .dc.b   2,0
110: aifont: .dc.b   "愛",00
```

リスト2

```
  1: *
  2: *        ラスターデモンストレーション 1
  3: *                                       波型ラスター
  4: *
  5: *        このプログラムではd6・d7をワークとして予約しています。
  6: *
  7: *        d6 = scrl_base_count
  8: *        d7 = scrl_index_count
  9: *
 10: *        カーソル ← →↑↓ = ラスター値変更           ESC = 終了
 11: *
 12:
 13:
 14:         .include        iocscall.mac
 15:         .include        doscall.mac
 16:
 17:         .xref           screen_init
 18:
 19: DI      macro
 20:         ori.w   #$0700,sr       * 割り込み禁止
 21:         .endm
 22: EI      macro
 23:         andi.w  #$f8ff,sr       * 割り込み許可
 24:         .endm
 25: MFP_IERB        equ     $e88009
 26:
 27: *
 28: *                       イニシャライズ
```

▶あの「ザナドゥ」の感動は過去への感傷なのだろうか？　　喜屋武　盛道(20)東京都

```
 29: *
 30: start:
 31:         sub.l   a1,a1
 32:         IOCS    _B_SUPER
 33:         moveq.l #$c,d1
 34:         IOCS    _CRTMOD         * 画面初期化
 35:         IOCS    _G_CLR_ON       *
 36:
 37:         jsr     screen_init
 38:
 39:         clr.l   d6
 40:         clr.l   d7
 41:
 42:         move.b  MFP_IERB,intdata * ちらつきの原因になる割り込みを禁止する
 43:         move.b  #%00001110,MFP_IERB * タイマC/D割り込み禁止
 44:
 45:         DI                      * 割り込み禁止
 46:         moveq.l #1,d1           * V-BLANK割り込み設定
 47:         lea.l   vblankint,a1
 48:         IOCS    _VDISPST
 49:         tst.l   d0
 50:         bne     quit
 51:
 52:         lea.l   hsyncint,a1     * H-SYNC割り込み設定
 53:         IOCS    _HSYNCST
 54:         tst.l   d0
 55:         bne     quit2
 56:         EI                      * 割り込み許可
 57: *
 58: *                  メイン処理
 59: *
 60: mainloop:
 61:         jsr     wave_data_set   * 次に表示するウェーブデータを作る
 62:
 63:         clr.w   d0
 64:         move.w  #0,d1
 65:         IOCS    _BITSNS         * ESCキーが押されるまでループ
 66:         cmp.b   #%0000_0010,d0
 67:         bne     mainloop
 68:
 69: break:  DI                      * エンド処理
 70:         suba.l  a1,a1
 71:         IOCS    _HSYNCST
 72: quit2:  suba.l  a1,a1
 73:         IOCS    _VDISPST
 74:         EI                      * 割り込みを再許可
 75: quit:
 76:         move.w  #0,-(sp)
 77:         DOS     _KFLUSH         * キーバッファをクリア
 78:         addq.l  #2,sp
 79:         move.w  #1,-(sp)
 80:         move.w  #16,-(sp)
 81:         DOS     _CONCTRL        * 画面を 512*768 に初期化
 82:         addq.l  #4,sp
 83:         move.b  intdata,MFP_IERB
 84:         DOS     _EXIT
 85:
 86: *
 87: *       割り込みルーチン
 88: *
 89: vblankint:              * V-BLANK割り込み
 90:         movem.l d0-d1,-(sp)
 91:         move.b  #1,b_blank      * 判定セット
 92:         clr.w   d0
 93:         move.w  #7,d1
 94:         IOCS    _BITSNS
 95:         move.b  d0,keydata
 96:         cmpi.b  #%0001_0000,d0  * カーソル上
 97:         beq     yscl_add
 98:         cmpi.b  #%0100_0000,d0  * カーソル下
 99:         beq     yscl_sub
100:         bra     vblankquit      * 何も押されなかった
101: yscl_add:
102:         addq.w  #8,d6
103:         andi.w  #%00000111_11111111,d6  * スクロールベースカウンタゴミ取り
104:         bra     vblankquit
105: yscl_sub:
106:         subq.w  #8,d6
107:         andi.w  #%00000111_11111111,d6  * スクロールベースカウンタゴミ取り
108: vblankquit:
109:         move.w  #0,d7           * スクロールインデックスカウンタ初期化
110:         movem.l (sp)+,d0-d1
111:         rte
112:
113: hsyncint:               * H-SYNC割り込み
114:         move.l  a0,-(sp)
115:         lea.l   wavedata,a0
116:         adda.w  d6,a0
117:         adda.w  d7,a0
118:         move.w  (a0),$eb0800    * BG横
119:         move.w  (a0),$eb0802    * BG縦
120:         addq.w  #4,d7
121:         move.l  (sp)+,a0
122:         rte
123: *
124: *       サブルーチン
125: *
126: *
127: *       ウェイブデータの生成
128: *
129: wave_data_set:
130:         movem.l d0-d2/a1-a6,-(sp)
131:         tst.b   b_blank         * V_BLANKの判定
132:         beq     wavequit        * V_DISP中に1回しか処理をやらない
133:         move.b  #0,b_blank      * 判定リセット
134:
135:         lea.l   sindata,a1
136:         lea.l   endsin-4,a2
137:         lea.l   wavedata+000*4,a3
138:         lea.l   wavedata+064*4,a4
139:         lea.l   wavedata+128*4,a5
140:         lea.l   wavedata+192*4,a6
141:         move.w  #64-1,d2
142:
143:         move.w  #7,d1
144:         move.b  keydata,d0
145:         cmpi.b  #%0000_1000,d0  * カーソル左
146:         beq     waveloop00
147:         cmpi.b  #%0010_0000,d0  * カーソル右
148:         beq     waveloop01
149:         bra     wavequit        * 何も押されなかった
150:
151: waveloop00:
152:         clr.l   d1
153:         move.w  (a1),d1
154:         asl.l   #3,d1
155:         addq.l  #8,a1
156:         add.l   d1,(a4)+
157:         sub.l   d1,(a6)+
158:         clr.l   d1
159:         move.w  (a2),d1
160:         asl.l   #3,d1
161:         subq.l  #8,a2
162:         add.l   d1,(a3)+
163:         sub.l   d1,(a5)+
164:         dbra    d2,waveloop00
165:         bra     wavecopy
166:
167: waveloop01:
168:         clr.l   d1
169:         move.w  (a1),d1
170:         asl.l   #3,d1
171:         addq.l  #8,a1
172:         sub.l   d1,(a4)+
173:         add.l   d1,(a6)+
174:         clr.l   d1
175:         move.w  (a2),d1
176:         asl.l   #3,d1
177:         subq.l  #8,a2
178:         sub.l   d1,(a3)+
179:         add.l   d1,(a5)+
180:         dbra    d2,waveloop01
181:
182: wavecopy:
183:         move.w  #256-1,d0
184:         lea.l   wavedata+000*4,a3
185:         lea.l   wavedata+256*4,a4
186:         lea.l   wavedata+512*4,a5
187:         lea.l   wavedata+768*4,a6
188: waveloop02:
189:         move.l  (a3),(a4)+
190:         move.l  (a3),(a5)+
191:         move.l  (a3)+,(a6)+
192:         dbra    d0,waveloop02
193: wavequit:
194:         movem.l (sp)+,d0-d2/a1-a6
195:         rts
196: *
197: *
198:         .data
199:         .even
200:
201: sindata:        *SINデータ2048倍 256個      0~90°
202:         .dc.w   $0800,$07FF,$07FF,$07FF,$07FF,$07FF,$07FE,$07FE
203:         .dc.w   $07FD,$07FC,$07FC,$07FB,$07FA,$07F9,$07F8,$07F7
204:         .dc.w   $07F6,$07F4,$07F3,$07F2,$07F0,$07EF,$07ED,$07EB
205:         .dc.w   $07E9,$07E7,$07E5,$07E3,$07E1,$07DF,$07DD,$07DB
206:         .dc.w   $07D8,$07D6,$07D3,$07D0,$07CE,$07CB,$07C8,$07C5
207:         .dc.w   $07C2,$07BF,$07BC,$07B9,$07B5,$07B2,$07AE,$07AB
208:         .dc.w   $07A7,$07A4,$07A0,$079C,$0798,$0794,$0790,$078C
209:         .dc.w   $0788,$0784,$077F,$077B,$0776,$0772,$076D,$0768
210:         .dc.w   $0764,$075F,$075A,$0755,$0750,$074B,$0745,$0740
211:         .dc.w   $073B,$0735,$0730,$072A,$0725,$071F,$0719,$0714
212:         .dc.w   $070E,$0708,$0702,$06FC,$06F5,$06EF,$06E9,$06E3
213:         .dc.w   $06DC,$06D6,$06CF,$06C8,$06C2,$06BB,$06B4,$06AD
214:         .dc.w   $06A6,$069F,$0698,$0691,$068A,$0683,$067B,$0674
215:         .dc.w   $066C,$0665,$065D,$0656,$064E,$0646,$063E,$0637
216:         .dc.w   $062F,$0627,$061F,$0616,$060E,$0606,$05FE,$05F5
217:         .dc.w   $05ED,$05E5,$05DC,$05D3,$05CB,$05C2,$05B9,$05B1
218:         .dc.w   $05A8,$059F,$0596,$058D,$0584,$057B,$0571,$0568
219:         .dc.w   $055F,$0556,$054C,$0543,$0539,$0530,$0526,$051C
220:         .dc.w   $0513,$0509,$04FF,$04F5,$04EB,$04E2,$04D8,$04CE
221:         .dc.w   $04C3,$04B9,$04AF,$04A5,$049B,$0490,$0486,$047C
222:         .dc.w   $0471,$0467,$045C,$0452,$0447,$043D,$0432,$0427
223:         .dc.w   $041C,$0412,$0407,$03FC,$03F1,$03E6,$03DB,$03D0
224:         .dc.w   $03C5,$03BA,$03AF,$03A4,$0398,$038D,$0382,$0376
225:         .dc.w   $036B,$0360,$0354,$0349,$033D,$0332,$0326,$031B
226:         .dc.w   $030F,$0304,$02F8,$02EC,$02E1,$02D5,$02C9,$02BD
227:         .dc.w   $02B1,$02A6,$029A,$028E,$0282,$0276,$026A,$025E
228:         .dc.w   $0252,$0246,$023A,$022E,$0222,$0216,$0209,$01FD
229:         .dc.w   $01F1,$01E5,$01D9,$01CC,$01C0,$01B4,$01A8,$019B
230:         .dc.w   $018F,$0183,$0176,$016A,$015E,$0151,$0145,$0138
231:         .dc.w   $012C,$0120,$0113,$0107,$00FA,$00EE,$00E1,$00D5
232:         .dc.w   $00C8,$00BC,$00AF,$00A3,$0096,$008A,$007D,$0071
233:         .dc.w   $0064,$0057,$004B,$003E,$0032,$0025,$0019,$000C
234: endsin:
235: intdata:
236:         .ds.b   1
237: b_blank:                        * V_BLANKの判定
238:         .ds.b   1
239: wavedata:                       * ラスタースクロールするためのウェーブデータ
240:         .ds.l   1024
241: keydata:                        * キーデータ
242:         .ds.b   1
```

▶先日，初めて“チョコビ”を食べました。味はチョコボールみたいです。
井上　理(17)埼玉県

リスト3

```
  1: *
  2: *         ラスターデモンストレーション 2
  3: *                                   地面に応用出来るラスター
  4: *
  5: *         このプログラムではd7をワークとして使用しています。
  6: *
  7: *         d7 = scrl_index_count
  8: *
  9: *         カーソル ← → = ラスター値変更        ESC = 終了
 10: *
 11:
 12:
 13:           .include        iocscall.mac
 14:           .include        doscall.mac
 15:
 16:           .xref           screen_init
 17:
 18: DI        macro
 19:           ori.w   #$0700,sr       * 割り込み禁止
 20:           .endm
 21: EI        macro
 22:           andi.w  #$f8ff,sr       * 割り込み許可
 23:           .endm
 24:
 25: MFP_IERB        equ     $e88009
 26: MFP_IMRA        equ     $e88013
 27: MAXSPEED        equ     2       * スクロールの最大速度
 28: STARTRAS        equ     200     * スクロール開始ラスター
 29: *STARTRAS       equ     000     * スクロール開始ラスター
 30: SCLVRAS         equ     180     * ラスタースクロール幅
 31: *SCLVRAS        equ     567     * ラスタースクロール幅
 32:                                 * 開始ラスター+スクロール幅が
 33:                                 * 568未満になるようにして下さい。
 34: MAXRAS          equ     568     * 最大ラスター数
 35:                                 * 31KHz = 568本   15KHz = 260本
 36: *
 37:
 38: *                       イニシャルプロセス
 39: *
 40: start:
 41:           moveq.l #$c,d1
 42:           IOCS    _CRTMOD         * 画面初期化
 43:           IOCS    _G_CLR_ON       *
 44:           sub.l   a1,a1
 45:           IOCS    _B_SUPER
 46:
 47:           jsr     screen_init     * BGの表示
 48:
 49:           clr.l   d7
 50:
 51:           jsr     jimen_data_gene * ラスタースクロールの基本データ作成
 52:
 53:           DI                      * 割り込み禁止
 54:
 55:           move.b  MFP_IERB,intdata * ちらつきの原因になる割り込みを禁止する
 56:           move.b  #%00001110,MFP_IERB * タイマC/D割り込み禁止
 57:
 58:           moveq.l #1,d1           * V-BLANK割り込み設定
 59:           lea.l   vblankint,a1
 60:           IOCS    _VDISPST
 61:           tst.l   d0
 62:           bne     quit
 63:
 64:           lea.l   hsyncint,a1     * H-SYNC割り込み設定
 65:           IOCS    _HSYNCST
 66:           tst.l   d0
 67:           bne     quit2
 68:
 69:           EI                      * 割り込み許可
 70: *
 71: *                       メインプロセス
 72: *
 73: mainloop:
 74:           jsr     jimen_data_set  * 次に表示するラスターデータを作る
 75:
 76:           clr.w   d0
 77:           move.w  #0,d1
 78:           IOCS    _BITSNS         * ESCキーが押されるまでループ
 79:           cmp.b   #%0000_0010,d0
 80:           bne     mainloop
 81:
 82: *
 83: *                       エンドプロセス
 84: *
 85: break:    DI
 86:           suba.l  a1,a1
 87:           IOCS    _HSYNCST
 88: quit2:    suba.l  a1,a1
 89:           IOCS    _VDISPST
 90: quit:
 91:           move.b  intdata,MFP_IERB
 92:           EI                      * 割り込みを再許可
 93:
 94:           move.w  #0,-(sp)
 95:           DOS     _KFLUSH         * キーバッファをクリア
 96:           addq.l  #2,sp
 97:           move.w  #1,-(sp)
 98:           move.w  #16,-(sp)
 99:           DOS     _CONCTRL        * 画面を 512*768 に初期化
100:           addq.l  #4,sp
101:           DOS     _EXIT
102:
103: *
104: *                       割り込みルーチン
105: *
106: vblankint:                      * V-BLANK割り込み
107:           movem.l d0-d1,-(sp)
108:           move.b  #1,b_blank      * 判定セット
109:           move.w  #7,d1
110:           IOCS    _BITSNS         * キーデータの読み込み
111:           move.b  d0,keydata
112:           move.w  #0,d7           * スクロールインデックスカウンタ初期化
113:           movem.l (sp)+,d0-d1
114:           rte
115:
116: hsyncint:                       * H-SYNC割り込み
117:           move.l  a1,-(sp)
118:           lea.l   sctldata,a1
119:           adda.w  d7,a1
120:           move.w  (a1),$eb0800    * BG横
121: *         move.w  (a1),$eb0802    * BG縦
122:           addq.w  #4,d7
123:           move.l  (sp)+,a1
124:           rte
125: *
126: *                       サブルーチン
127: *
128: *
129: *         基本スクロール値（固定小数点演算用）データ生成
130: *
131: jimen_data_gene:
132:           movem.l d0-d2/a1,-(sp)
133:           clr.l   d0
134:           clr.l   d1
135:           lea.l   jimendata,a1
136: *         move.w  #MAXSPEED,d0            * MAXSPEEDを256で割る
137: *         swap    d0                      *
138: *         asr.l   #8,d0                   *
139:           move.w  #MAXSPEED*256,d0 * MAXSPEEDを256で割る
140:           move.w  d0,d1
141:           move.w  #SCLVRAS-1,d2
142: jimen_gene00:
143:           move.l  d0,(a1)+
144:           add.l   d1,d0
145:           dbra    d2,jimen_gene00
146:           movem.l (sp)+,d0-d2/a1
147:           rts
148: *
149: *         スクロールデータの生成
150: *
151: jimen_data_set:
152:           movem.l d0-d3/a1-a2,-(sp)
153:           tst.b   b_blank         * V_BLANKの判定
154:           beq     rasquit         * V_DISP中に1回しか処理をやらない
155:           move.b  #0,b_blank      * 判定リセット
156:
157:           lea.l   jimendata,a1
158:           lea.l   sctldata+STARTRAS*4,a2
159:           move.w  #SCLVRAS-1,d2
160:
161:           move.b  keydata,d0
162:           cmpi.b  #%0000_1000,d0  * カーソル左
163:           beq     rasloop00
164:           cmpi.b  #%0010_0000,d0  * カーソル右
165:           beq     rasloop01
166:           bra     rasquit         * 何も押されなかった
167:
168: rasloop00:
169:           move.l  (a1)+,d3
170:           add.l   d3,(a2)+
171:           dbra    d2,rasloop00
172:           bra     rassita
173: rasloop01:
174:           move.l  (a1)+,d3
175:           sub.l   d3,(a2)+
176:           dbra    d2,rasloop01
177:
178: rassita:
179:           move.l  -4(a2),d3
180:           move.w  #MAXRAS-(STARTRAS+SCLVRAS)-1,d2
181: rassita00:
182:           move.l  d3,(a2)+
183:           dbra    d2,rassita00
184: rasquit:
185:           movem.l (sp)+,d0-d3/a1-a2
186:           rts
187: *
188: *
189:           .data
190:           .even
191: intdata:                        * 割り込み情報退避ワーク
192:           .ds.b   1
193: b_blank:                        * V_BLANKの判定
194:           .ds.b   1
195: jimendata:                      * 加減算するためのデータ
196:           .ds.l   568
197: sctldata:                       * ラスタースクロールデータ
198:           .ds.l   568
199: keydata:                        * キーデータ
200:           .ds.b   1
```

こちらシステムX探偵事務所

FILE-Ⅷ

より質の高いモーフィングを目指して

ドローネ三角形分割の導入

Shibata Atsushi 柴田 淳

今回は先月号で春香嬢が持ってきた「ドローネ三角形分割」を利用してモーフィング実験を行っています。さあ，この前のモーフィング実験は9月号で行いましたが，今度はどんな仕上がりになったでしょうか？　楽しみですね。

illustration : T. Takahashi

柴田淳(以下Ats)：先月号の最後に，Macintoshのモーフィングソフトの話をしたのを覚えてますか？

マスター(以下M)：ああ，そのソフトでも実は画像の変形に三角形の自由変形を使っているらしい，とかいう話でしたよね。それで，今月はその証拠を見せるって。

琴張護（以下護）：しかしこともあろうにMacintoshのソフトが，そんな安っぽいことをしているものでしょうか。

Ats：そう思う気持ちもわからないでもないけど，Macintoshのソフトだからってなんでも高級なことをしていると思い込むのは間違いですよ。

琴張春香(以下春)：ふーん，そういうものかしらね。

M：ところで，その証拠の件はどうなったんですか。

Ats：それがですね，このソフトを意図的に暴走させてみたら，面白い画像を出力したんです。

護：意図的に暴走させたというと？

Ats：Macintoshのシステムでは，アプリケーションごとにどれくらいのメモリを割り当てるかが決められるんですけどね，そのソフトへのメモリの割り当てを，規定の最小値より少なめに取ってみたんです。すると，写真1のような画面を出力したんですよ。

春：この写真1，モナリザみたいに見えるけど。

M：なんだかハサミで切り刻まれたようなというか，変な出力ですね。

Ats：このソフトには「カリカチュア」という機能があるんです。要するに2つの画像をオーバーラップさせずに，変形のみを行う機能なんですけど，その最中にアプリケーションを暴走させて得た画像なんです。

護：それにしても，どうしてこんな画像を出力するのでしょう。

Ats：ええと，アプリケーションが立ち上がるとき，システムがメモリを割り当てるわけでしょ。割り当てられた領域の下位にプログラム本体が，上位にはアプリケーション専用のスタックが割り振られます。で，そのスタックとプログラムのあいだがデータなどを置いておく領域になるわけですよね。

護：なるほど，ではこの場合は規定より少なくメモリが割り当てられているので，アプリケーションを走らせているうちにスタックが下りてきて，データ領域を侵食してしまうというわけですね。

Ats：そうなんですよ。でね，おそらくデータ領域には12月号のような方法で出力された三角形分割の座標情報などがしまい込まれていると思います。そして，その情報が書き換わるものだから，このような出力になってしまうというわけ。

M：すると，この画像は変形後の三角形が，本来張りつけられるべき場所に描かれないために，こんな切り刻んだような画像になったということですか。

Ats：推測ですけどね。

護：たしかにここにある写真によって，このソフトでは変形前に画像を三角形に分割しているということはわかるかもしれません。しかし，分割後に三角形の自由変形を使って画像を変形させているということは証明できていないのではないでしょうか。

Ats：そうくると思って，こういう写真も用意してきました。

春：写真2ね。これはどういう写真なの？

Ats：その話をする前に，このソフトの仕組みについて説明しましょう。2つの画像を滑らかに補間するとしますよね。

M：一般にいうモーフィングを行う場合ですね。

Ats：まず最初に，2つの画像の対応する点，たとえば目尻や口元とかに点を打っていくんです。

護：多角形で囲むわけではないのですね。

Ats：そうなんですよ。ただし，それに近いことはできます。打った点を，線で結びつけることができるんです。このソフトの以前のバージョンでは，この線は単に視認性を高めるだけの役割しか持っていず，画像の出力には関係がなかったみたいです。

春：点を打ったあとはどうするの？

Ats：点を打ったあとに，メニューからモーフという項目を選ぶんです。これを行わないと，画像の変形にとりかかれないんです。そして，打った点の数に比例してこの

写真1　ハサミで切り刻まれたようなモナリザ

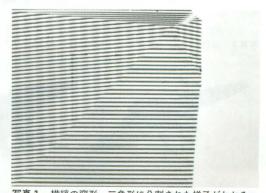



写真2　横縞の変形。三角形に分割された様子がわかる

作業にかかる時間が長くなっていきます。以上のことを考慮すると，この画像の上に打たれた点をもとに，三角形分割を行っているんだと思うんですけどね。

M：すると，そのモーフという作業が終わってから，晴れて画像を出力できるというわけですね。

Ats：あと，点と点を線で結びつけると，必ずその線が分割線として選ばれるようになっているようです。その証拠に，この線をわざと交差させると，分割が破綻して出力がおかしくなるんですよ。

春：これでこのソフトの仕組みはわかったけど……。

Ats：ああ，そうそう，写真2ね。これは，横縞をこのソフトで変形させたものなんです。で，もしこのソフトが三角形の自由変形を使っていなくて，曲線的な方法を用いた変形を行っているとしたら，この横縞はグニャリと曲がるはずじゃないですか。

M：なるほど。でも，この写真を見る限り，縞模様は直線のままですね。

Ats：そうなんです。つまり，この写真を見せてなにがいいたいかというと，いままでやってきた，三角形分割と三角形の自由変形によってモーフィングを行えば，このソフト程度の出力は得られる，ということなんです。

出力画像の質を高める

Ats：さて，前回のモーフィング特集から，「誤差の少ない三角形自由変形」や「ドローネ三角形分割」を取り上げてきました。そこでこれからは，この2つを使ってもう少しきれいなモーフィング画像を作るにはどうするかを説明していきましょう。

春：ところで，前にやったときの問題点ってどんなのがあったかしら。

護：そうです。その問題点がいくらかでも解決されているのでなければ，今回モーフィングを取り上げる意味がなくなってしまいます。

Ats：ええと，問題点としては，

1) 変形時に誤差が出る。

2) 多角形の相互三角形分割に破綻が生じるときがある。

3) 変形時，多角形で指定した領域が重なって出力される。

というものがありました。

M：このなかの1)は，11月号で取り上げた誤差の少ない三角形の自由変形ルーチンを使うことで解消されますよね。

Ats：今回使っている三角形自由変形ルーチンは，さらに誤差が少なくなるよう改良を施したものですから，その点はバッチリです。

春：この2)はどういう意味？

護：モーフィングというのは，ひとつの図形からもうひとつの図形に連続的に変化していく画像を得るためのものです。そのためには，2つの画像の特徴的な部分，たとえば鼻や口などを，2つの画像上で同じように多角形で囲っていくのです。

Ats：相互三角形分割というのは，その際指定した多角形を，三角形自由変形で変形するために三角形で分割していく処理のことです。で，入力した多角形の形によっては三角形分割が正しく行われない場合があるわけです。

M：正しい分割にならないというと？

Ats：たとえば，分割線が交差してしまったり，多角形の外側に出てしまったり，という問題が起こります。

護：この問題を解決するために，新しい三角形分割のアルゴリズムであるドローネ三角形分割を導入するというわけですね。

Ats：ドローネ三角形分割を導入するについては，実はもうひとつ理由があるんです。これは3)に関係あることなんですが。

春：ところで，「多角形で指定した領域が重なる」ってどういうこと？

Ats：たとえば，人間の顔を変形させる場合を考えましょうか。顔の輪郭を多角形で囲って，そのあと目や鼻を囲うでしょう。

M：同じ作業を，変形元と変形先の画像について行うわけですね。そして，入力された多角形を三角形に分割すると……。

Ats：その三角形分割をもとにして，補間途中の画像を得るわけです。まず顔を，変形画像に三角形単位で切り張りして，そのあと目や鼻などを切り張りしていく。でも，顔を切り張りする段階で，すでに目や鼻などの顔のパーツも変形を受けているわけですよね。

春：そうだけど，なにか問題があるの？

護：顔を変形するときに張り付けられる目が，目を囲んだ多角形を張り付ける位置と同じ場所に現れるとは限らないというのが問題のような気がしますが。

Ats：そうなんですよ。つまり，変形中の画像に，同じものが重なって表示される場合があるんですね。

春：なるほど。でも，それとドローネ三角形分割とどういう関係があるわけ？

Ats：同じものが重なってしまう原因はひとえに，ある多角形のなかに完全に含まれる部分を，別の多角形として切り出してし

写真3　以前の方法。ゴミが出ているのがわかるだろうか？

写真4　今回の方法。変形誤差，相互分割によるゴミなどが取り除かれている

まうことにあるんです。。
M：でも，入力された多角形を三角形に分割することには変わりがないわけですから，ドローネ三角形分割を使っても，同じじゃないですか？
Ats：マスターのいうとおり，ドローネ三角形分割を使っても，同じように多角形を分割するのであれば同じような不具合が発生します。でも，多角形の分割の方法を変えれば，話は別でしょう？
春：多角形の分割の方法を変えるって，どういうこと？
Ats：ドローネ三角形分割のウマミは「領域内に散りばめられた点を結んで，領域の三角形分割を行う」というところにあるんです。つまり，指定された多角形のすべての頂点を入力点とみなして，分割をすればいいじゃないですか。
護：そうですね。この方法だと，変形中に同じものが重なって現れることはなくなりそうです。
Ats：つまり，以前の方法だと，ひとつの画像を何枚かの多角形として処理してたんですけど，ドローネ三角形分割を使うと，ひとつの画像をひとつの多角形として処理することができるというわけです。

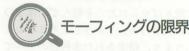
前回の出力との比較

Ats：こういう類のものは，理屈でどうこういうより現物を見てもらったほうが話は早いですよね。で，以前の方法（写真3）と今回の方法（写真4）で出力したモーフィング画像を比較してみることにしましょう。これはどちらも，同じ多角形で変形させたものです。
護：写真4では三角形自由変形の誤差を取り除いた効果が，かなり出ているように思えますが。
M：以前のように，明らかに「ゴミ」だとわかる部分がなくなってますね。
Ats：入力された多角形を分割する際，破綻が起こらなくなったというのも，ゴミ軽減に多少貢献しているかもしれません。
M：三角形分割が破綻しているときは，本来なら描かれないはずの場所に三角形が描かれてしまうわけですからね。
春：ドローネ三角形分割の，領域内の点を結んで三角形分割を行うという方法のほうが，モーフィングにはやはり適しているみたいね。
Ats：それに，変形の原理さえ頭に入れておけば，多角形の切り出し方を工夫することで，もっと質の高い画像を出力することもできます。
M：たとえばどんなふうにするんですか？
Ats：入力した多角形の頂点を点集合とみて，その隣り合った点同士を結んで三角形にするわけですから，そのアルゴリズムを見越して，画面を細かく分割してくれるように多角形を切り出すんです。
護：端的にいえば，入力する多角形の数を増やすということではないでしょうか。
M：なるほど。画面が細かく分割されれば，出力の質が高くなるというのは直感的にうなずけますね。
Ats：ええと，もっと違いがわかりやすいのがこちらの連続変形，写真5，6でしょう。また僕の顔写真で恐縮なんですが。
春：これは2つの画像を補間するんじゃあなく，ただ変形させただけの写真ね。
Ats：そうなんです。2つの画像のあいだを連続的に補間する，ごく普通にいうモーフィングの場合，2つの画像を重ねる時点でオーバーラップという画像処理をするんです。
護：2つの画像を，色の平均を取ることで合成する方法ですね。
Ats：で，そのオーバーラップを行うと変形で得られた画像に出たゴミだとかが，見えにくくなってしまうんですよね。だから，オーバーラップをしない変形だけの比較もしてみたんです。
M：さっきいっていたとおり，以前までの方法で変形を行った写真では目などの顔のパーツが重なって現れていますね。
護：でもこの変形は，以前の方法ではうまく変形できないような画像を，故意に取り上げたように思えますが。
Ats：たしかにそうです。でも逆に，これはいままでうまくいかなかったことが解決されている，という証明にもなっていると思います。

写真5　以前の方法。鼻の上に口が重なっている

写真6　今回の方法。ドローネ三角形分割を利用

Ats：ところで，これまでは出力された画像を写真で見せるだけでしたが，ここで少し趣向を変えて，どんなふうにしてモーフィング画像を出力するまでに至るか，そのプロセスを少しお見せしましょう。
M：ほほう，それは面白そうですね。
Ats：まず，変形させる素材の調達が第1段階です。これは写真を撮ってきてスキャナで取り込むなり，あと個人的な利用であればテレビからタレントの顔を撮り込むなんてことも考えられますね。
春：写真7が，これから変形させようとする画像ね。あれ，なんだか画像に加工をしているみたいだけど。
Ats：変形に備えて，余分なところを削り取っています。
M：余計な部分って，キリンとダチョウの背景にある森も消すんですか？
Ats：そうなんです。この写真7にある遠景の森のような，複雑というか不規則な画像は，うまく変形できないんですよ。
護：では，CGAに使うとき，バックに森があるカットが必要な場合などはどうするのでしょう。
Ats：その場合は，別途に森の画像を用意して合成するんです。プロ用の機材を使ってモーフィングをするときでも，このような手順を踏むことには変わりありません。
春：へえ，そうだったの。
Ats：変形させたい素材の切り出しが終わったら，次に画像分割に利用する多角形を切り出す作業に入ります。
M：あれ？　なんですか，このソフトは。
Ats：ああ，この写真8に写っているソフトは，恥ずかしながら僕の自作のものなんです。まだ作っている途中で，素材を読み込んで多角形を分割する機能しかありませんけどね。
護：将来的には，必要な機能をすべてこのソフトに集約することになるのでしょう？
Ats：当然です。で，多角形の切り出しが終わったら情報をファイルに書き出して，その情報を元に三角形分割を行います。その情報をまたファイルに落として，モーフィングを行うドライバを起動して画像をPIC形式で出力する，というのが現時点での作業の流れです。ちなみに，写真9では見やすさを考慮して，背景を描き足しています。
M：画像はPICで出力するんですか？　じゃあ，これをアニメーションとして見られないじゃないですか。
Ats：動画として見るためには，DōGAシステムのお世話になる方法がオーソドックスでしょう。システムの中にグラフィックの指定領域をDōGAのPIC形式で保存する外部コマンドがありますので，1枚ずつPICで表示してはDōGA形式に変換するバッチファイルを書いて，あとはDōGAシステムのHANIM.Xを起動すればアニメとしてモーフィング画像を見ることができます。
春：ふーん，意外と面倒くさいのね。
Ats：でも，アニメーションを作ろうと思ったらこれくらいの覚悟はしないと。

モーフィングの限界

Ats：ところで，モーフィングといえばちょっと前に，マサル君からラモスに変わっていくCFが，評判になりましたよね。
春：ほかにはコーヒーのCFに出てくる柴田恭兵とかもあるわよね。
M：でも，あれはちょっとムリがあるような気がするな。
Ats：コーヒーのCFはどうでもいいんですけど，あのラモスのCFってどうやって撮ったか知ってます？
護：まず，カメラの動きはモーションコントロールカメラかなにかで制御しているのでしょう。
M：あと，変形途中でラモスが横を向くじゃないですか。あのタイミングなんかも，合わせないとまずいですよね。
春：つまり，人間対人間の変形の場合，変形前と変形後のモデルのカメラとの相対位置，顔の向きなどを，合わせておかなけ

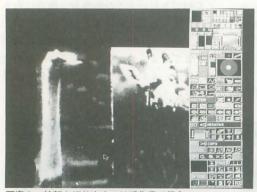



写真7　輪郭を切り出すのは手作業で行う

写真8　自作の，多角形を切り出すツール

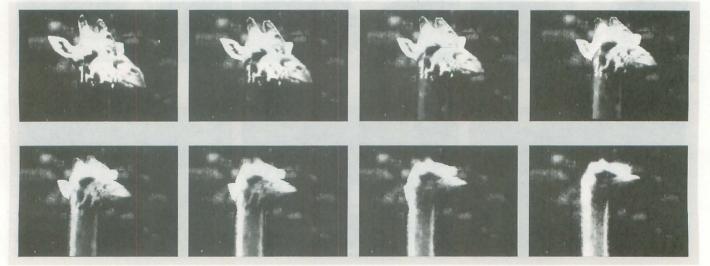



写真9　キリンからダチョウに変形していく連続写真。目や口などを多角形として切り出した

ればならないってことね。
Ats：あのCFのもう少し細かい部分についても考えてみましょうか。たとえば，変形前のマサオ君の髪型を覚えていますか？
M：長髪じゃあなかったはずですよね。
春：なんか，角刈りに近いほど短かったような気がするけど。
Ats：そう，たしか相当短い髪型でしたよね。それに対して，ラモスはもじゃもじゃの長髪じゃないですか。モーフィングで画像を変形させる場合，こんなふうに大きく異なる2つの素材を取り扱うのってけっこう大変なんですよ。
護：しかし，長髪であっても短髪であっても髪の毛であることに変わりはないのですから，2人の髪の毛を多角形で囲ってしまえば，変形はできるでしょう。
Ats：いや，僕がいいたかったのはできるかできないかの問題じゃなくて，質を高く仕上げるにはどうすればいいかってことなんです。たしかに琴張さんのいった方法で変形できなくはありませんけど，それでは髪の毛が出現するような画像になるじゃないですか。
護：なるほど。もっともな話です。
Ats：あのCFをよく見るとわかるんですけど，変形途中にラモスの髪はパサリと落ちるようにのびていくんですよ。
春：ふーん，どうやったらそうなるのかしら。
Ats：2つの方法があると思います。まず1つ目は，実際にラモスの髪を落とす方法。最初，天糸かなにかで束ねておいて，ころあいを見計らってその糸をほどくみたいな方法です。
M：ハイテクにローテクを忍ばせるんですね。
Ats：2つ目の方法は，変形を行う際，変形する先の座標情報を綿密に設定して，モーフィングによって落ちているように見せる方法。この場合，髪の毛のような細い線が束になった，モーフィングがあまり得意としない素材を扱うわけですから，けっこう大変でしょうね。
春：で，柴田君はどっちだと思うわけ？
Ats：僕はたぶん1つ目のほうだと思うんですけど，あまり自信はありません。
護：それにしても，このように細かく掘り下げていくと，モーフィングも画像変形の手段としては万能でないという気がしてきました。
Ats：その認識は，かなり重要だと思うな。モーフィングというのは，たしかに柔軟ではあるけれども，あくまでも画像変形の手段でしかないわけですよね。
M：そういえば，矩形領域の画像を小さくする処理に，モーフィングを使おうとする人はまずいないですものね。
Ats：つまり，まずなにかをしようという目的があって，その目的を達成しようとする手段がある。そしてモーフィングは，目的でなく手段でしかないんです。モーフィングという画像変形の方法があると知って，これを使ってなにかをやってみようというアプローチはたしかにあるけれど，そのような方法論からは本当にモーフィングを使いこなしたような作品は生まれないと思うんです。

M：そういえば，最近はハリウッドの映画でもモーフィングをあからさまに使った演出は見かけないですよね。
護：ロバート・ゼメキス監督の「永遠に美しく」では，不老不死を手に入れた登場人物の傷が治っていくシーンなどに，さりげなく使われていました。
Ats：あと，もっとメジャーなところでは「ジュラシックパーク」の恐竜が歩くシーンで，筋肉の表現に使われていたりと，指摘されなければ気づかないような使われ方が増えているようです。
春：ふーん，「ターミネーター2」の頃はあれだけ騒がれたモーフィングなのにね。
Ats：とにかく誤解してほしくないのは，モーフィングを使うこと自体には，ほとんどなんの価値もないんだということです。肝心なのは，モーフィングを使ってなにをするか，という目的のほうなんですね。また，モーフィングのようなけっこう複雑な画像処理を用いるとき，その仕組みを知っているのと知らないのでは大きな差があることもたしかです。
春：じゃあ，私みたいに原理をよく理解できない人は，モーフィングのような道具を使うべきじゃないってこと？
Ats：いや，なにも全部知っている必要はないと思うんです。でも，サワリだけでも知っているとかなり有利だと思うんですよ。モーフィングに限らず，車やバイクなども，これと似たような要素を持っているんじゃないですかね。
M：さて，今回でモーフィング実現へのアプローチも，ひと区切りついたって感じですね。これからはどうしますか？
Ats：区切りのいいところでいったん止まって，あとはいままでの成果を皆さんに見てもらうために体裁を整える作業に入ろうかと思います。ただ，今後うまい改良の指針が見えてくればより質の高いモーフィングを目指してアプローチを始めるかもしれないし，そのあたりはなんともいえません。
護：ほかにやることがあるのですか？
Ats：ネタはいくつかありますよ。ただそういうことのほかに，最近やたらとやることが増えてきて，そのくせやりたいこともいっぱいあって，皆さんには悪いんですけどしばらくモーフィングから解放されたいんです。
春：じゃあとにかく，今後の予定は未定ということね。
Ats：だいいち，僕がやるかどうかもまだわかりませんよ。そんな状態なのに予定もくそもあったもんじゃない。
護：なにやら，だんだん口調が投げやりになってきたような気が……。
Ats：ううっ，ちくしょう。今月これで1週間近く睡眠時間3時間の日が続いてるんだ。僕は10時間寝なくちゃ体がきちんと働かない体質なのに，この状態で来月のこと考えろったって無理に決まってるじゃないか。
春：まあまあ，そう興奮しないで。
M：柴田君，どうしたんですか，急に。
護：なにか，鬱積したものが一気に放出されている感じです。
Ats：ぼ，僕は童顔だけど，これでけっこう年をとっているんだぞ。最近は立ち上がるときに無意識に「よっこらしょ」って……。

（つづく）

(で)のショートプロぱーてぃ――その52

オトナのためのショートプロ!?

Komura Satoshi 古村 聡

今月はゲーム1本にツールが2本。プログラムはX-BASICにC言語，アセンブラと盛りだくさん。ちょっとシブメのショートプロです。ついに（で）氏もオトナの魅力を身につけたのでしょうか？ コラムでは，ぷろぐらむ風まかせがスタートです。

illustration：T.Takahashi

どうもどうも(で)でございます。いやー，私もついにオトナの世界に足を一歩踏み入れてしまいました。シブさの帝王，和製ハンフリー・ボガードと呼んでくれたまえ。

ほら，よくドラマとかであるじゃないですか，行きつけの店ってやつ。ちょっと薄暗いカウンターでコートを脱ぎつつ「いつものやつね」なんていうと，バーボンなんかがグラスに入って，ゴトッと置かれるの。くー，かっこいいよなー，シブイよなー。ずっとあこがれていたんです，あーゆうのに，私。

で，私にもできたんですよ，行きつけの店が。線路のガード下にある小さな古い店なんです。冬の寒い日に白い息を吐きながらガラガラっと今にも壊れそうな戸を開けると，そこには電車が来るとガタッガタッと揺れるちょっと煤けたカウンターがあったりするんです。でね，おもむろに重いカーキ色のコートを脱ぎながら，椅子に座って「いつものヤツね」っていうと出てくるようになったんです。

ただし，出てくるのはお酒じゃなくて「てんぷらうどん生卵，わかめ入り」なんだけど(だって，お店ってガード下の酒場じゃなくてその隣のそば屋さんなんだもん)。

いや〜，俺ってオ・ト・ナ。今日もうどんが，んまいぜ。

オトナの香りのパズルゲームだ

ということで今月はショートプロもオトナの世界だったりします。今月の1本目のプログラムは茨城の荒井さん作のオトナの香りただよう知的なゲーム，ALP.Cです。どうぞっ。

ALP.C for X680x0(要Cコンパイラ)

茨城県 荒井一光

このプログラムはC言語で書かれたちょっとかる〜い，パズルゲームです。まずはリスト1をエディタで入力してください。セーブしたあと，CコンパイラPRO-68K ver.2.0以上かGCCを使って実行ファイルを作ります。それを実行すると，たとえば，

A G D F E C B

* * * * * * * *

なんてランダムにA, B, C, D, E, F, Gのアルファベットとアスタリスクが表示されます。これをAからGのアルファベット順に並べ替えるのがこのプログラムの目的です。

しかし，ただ並べ替えるだけじゃあ，面白くない。てなことで，このゲームではアルファベットを並べ替えるのにルールがあるのです。

ルール：画面上で連続した3文字を1セットとして，ほかのアルファベット間に挿入する。

つまり連続したアルファベット3文字を一度に動かさなくてはいけないのです。このゲームの入力装置はマウスオンリーなんですが，まず，移動させるアルファベット(連続した3文字)が決まったらいちばん左のアルファベットをクリックしてください。すると，移動させるアルファベットに赤でマークがつきます。もし間違って指定したときには右クリックしてください。マウスカーソルはどこでもOKです。

次に'*'を選びます。挿入したいところにマウスカーソルを動かして，マウスの左ボタンをクリックします。さっき選んだ3文字がそこに入って，文字を並べ替えることができるってわけです。

これを繰り返してABCDEFGと並べ替えられたら無事クリアです。また，文字の下に表示されている「見」「戻」「終」ですが，これらを左クリックすることにより，

見：コンピュータが問題を作ったときの過程を表示します。ギブアップのとき見てください。

戻：同じ問題を初めからやり直せます。いわゆるハマリの状態のときクリックしてください。

終：ゲームを終了します。途中であきたらクリックしてください。

以上の操作ができます。

なかなか知的な香りのするゲームなんであります。熱中するにはちょっとものたりないですけど，やっぱりこういう手軽な息抜きのゲームも必要ですよね。コンピュータも先のルールに従ってABCDEFGという並びから問題を作っているのでどの問題も絶対に解けるはずです。

オトナに必要なのはお仕事。でも，お仕事の合間にちょっとブレイク，かるーく頭をほぐしてまたお仕事。うーん，オトナ(なんとかWindowsの宣伝みたい)。冗談はともかく，最近，SX-WINDOWを使うことが多いのでこいつがSX上にあったらなー，という気がしてなりませんです，はい。だれか移植しません？

ちなみにこのゲーム，なにもプログラムに手を加えないと，コンピュータは問題を作るのに，A〜Gまで順番に並んだアルファベットの順番を2回しか並べ替えをしません。したがって，どの問題も解こうと思えば最短2手で解けます。ゲームをやりこんで解くのが簡単になってきたら，この並べ替え回数を変更してみるなどして遊んでくださいね。ちなみにこの並べ替え回数はdefineでMIXに定義してあります。

オトナはグラフも描くのだ

続きましてはっと。おおっと，なんだか今度もオトナの香りのするプログラム……かな？ 今月の2本目のプログラムは千葉の砂原さんのプログラム，最小二乗法を使った推測グラフ描画ツール（？）SIGMA.BASです。

SIGMA.BAS for X680x0(X-BASIC)

千葉県 砂原弘幸

このプログラムはX-BASIC用のプログ

ラムなんですが，プログラムを実行する前に準備が必要になります。

まず，最初にカレントディレクトリの下に，実測値を書いたファイルを入れるためにDATAというディレクトリを作ってください。コマンドライン上からであれば

A>mkdir data

で作ることができます。それが無事終わったら，リスト2をBASICのエディタから入力して，RUNしてください。プログラムが動きます。

えーっと，このプログラムはですね，最小二乗法により実測値(実験値)より推測される曲線や直線を求めるプログラムだったりするのです。なんのことかわかるかな。簡単にいうとなにか実験などをして，測ったいくつかの数値を入れると数値と数値のあいだを推測して適当なグラフを作ってくれるプログラムなのです。あ，最小二乗法ってなーにとかいわないようにね。私も知らないから(っておい)。いーんだよ，使い方さえわかれば。それがオトナってものさ。

ま，それはともかく，RUNしたあとは画面に選択肢が出ていますので，指示通りにテンキーから数字を打ち込んでいってください。入力ミスはプログラムで極力対応できるようになっていますが完全ではありません。できるだけミスのないように入力してください。また，実測値がマイナスの場合やグラフの点の位置が極端におかしな場合はエラーがおきて止まってしまいますので注意してください。

入力し終わると，グラフィック画面上にその実測値と実測値を適当につないだ結果のグラフが表示されます。

このプログラムは基本的に，

$$y=\sum_{i=1}^{n}\alpha i x^{Mi} \quad (Miは任意)$$

の形式にしか対応していません。このMiとαiを求めるプログラムです。たとえばグラフが$y=a0x^0+a1x^1+a2x^2$のような形式のようだな～と思ったときは展開項の数は3項で指数部分は0，1，2となるわけです。この予測される計算式の指数部分は自分で指定することになります。

っていうことなんですけど，みなさんわかりましたか？　いや～，凄いですねぇ。なんだか昔，理科の実験で適当に手を抜いてやってグラフをごまかしてそれっぽく描いたのを彷彿とさせますねー。えっ，そういうこととは違うんですか？(笑)

もともと，作者の砂原さんはこのプログラムを実験のデータ解析用に作ったんだそうです。うーん，深いよなぁ。なんだか，よくわからないや。オトナの世界って凄い。いかん，私がオトナなんだったけか。すっかり忘れてた。

あ，そうそう，データを入れるディレクトリ名が気に入らない場合にはプログラム中のグローバル変数dirという変数に入っていますので適当にプログラム中の変数名とディレクトリ名の両方を変えてください。また，データファイルの拡張子も.DATという名前で固定されていますが，これもkakuという変数に入っています。

プログラムは必要に応じて書き換える，そして使う，これがショートプロする人，ショートプラー（？）の道なんであります。うんうん。深い（と，深くツッコムのをさける(で)であった)。

X680x0の限界へ挑戦！

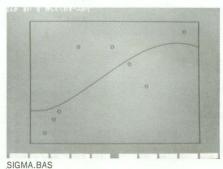
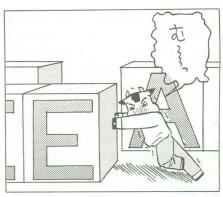
んでもって，今月トリのプログラム。兵庫県の曽谷さん作のスプライト256個を同時に表示するプログラムSP_ALL.Xです。どうぞ。

SP_ALL.X for X680x0
(要アセンブラ，リンカ，マクロファイル)
兵庫県　曽谷修二

我らがX680x0では，スプライトのパターンは256個まで定義できるものの，残念ながらそのスプライトを画面表示するときには1画面上に最大128個までしか表示できません。なぜならハードウェアにスプライトを表示するためのレジスタが128個分しかなかったりするからです。しかし，そこはなんでもありのショートプロ。このプログラムではスプライトの表示中にラスター割り込みのテクニックを使ってレジスタを書き換えてやって256個全てのスプライトを一度に表示しちゃうのです。

このプログラムはアセンブラのソースの形で書かれています。実行するための実行ファイルSP_ALL.Xを作るには，まず例のごとくエディタでリスト3を打ち込みます。それからコマンドライン上から

A>AS SP_ALL.S

SIGMA.BAS

A>LK SP_ALL.O

と打ってアセンブル，リンク作業をしてください。これで実行ファイルができます。あ，このプログラムは「X68000Develop.」などに収録されているアセンブラ&リンカHAS,HLKでも実行ファイルを作れます。それからのアセンブル，リンク作業にはDOSCALL.MACというマクロファイルが必要ですので気をつけてくださいね（もっともDOSCALLの中でも使っているのはEXITだけなのでちょっと力のある人ならちょいちょい，っと書き換えちゃうから，いらないかもしれないですけど)。

さて，実行ファイルができたら，まず，ゲームかなにかを立ち上げて少しだけ遊んでください。それから，

A>SP_ALL

とするとすべてのスプライトパターンがスプライトパレット0の色で画面に表示されます。「M」「,」「.」「/」「__」「スペース」「HOME」「DEL」のどれか(キーコードグループ6のキーですね)でパレット番号をひとつ増やした状態でスプライトを見ることができます（15番の次は0に戻ります)。プログラムを終了するには，「ESC」フルキーの「1」～「6」のどれかを押してください。

ああ，なんだか，今月は堅かったから「スプライト」なんて単語を見るとほっとする……。はっ，いかんいかん。私はオトナなんだ，うん。

久々の割り込み関係を駆使したテクニカル派のショートプロなんであります。やり

SP_ALL.X

ますね，作者の曽谷さん。このリストは要所要所で注釈も入っているから，「割り込み関係ってわかんなーい」といってるビギナープログラマの参考用にもいいですよね。ちなみにソース注釈の，「(512)」となっているところ（計6カ所）を指示どおりに変更すると512ドットモードで動いてくれるようにもなります。当然のようですけど，ラスター割り込みを使っている常駐ソフトとは一緒に使用しないでください。あまりないと思うけど，ねこみたいな，画面をいたずらしてまわる常駐ソフトなんかには注意してくださいね。ところでこのプログラムはフリーウェアとするそうです。

いや～，X680x0の限界を超えるプログラムなんでありますねー。試しにこのプログラムを実行しながらおもむろに電源を切ってみましょう。そのとき表示されているのがX68000が本来表示できるスプライトの限界なのです。でも，こんなことができてしまうと「うぉー，プログラミングテクニックがある限り，X68000の性能は無限なのだぁー！」と夕日の海に吠えてしまいたくなりますよねっ！

……いかん，もうだめだ。やっぱりこういうプログラムを見ると血が騒ぐ。オトナ気ないけどわくわくしちゃうんですよ。

ええい，もうオトナなんかやめっ！　いいんだいいんだ，わしなんかこーなったらコドモでもいいもん。遊んで遊んで遊んでやるんだい。てなことで，ぱーっと飲みにいくぞ。矛盾してるようだけどこれでいいのだ。また来月。

リスト1　ALP.C

```
  1: #include <stdio.h>
  2: #include <basic.h>
  3: #include <basic0.h>
  4: #include <mouse.h>
  5: #include <graph.h>
  6: #define MIX 2
  7: void randam();
  8: void hantei();
  9: void sort();
 10: int input();
 11:     struct A{
 12:         int no;
 13:         char moji;
 14:     }a[7],*b[7],*c,*d[MIX+1][7];
 15:
 16: void main(void)
 17: {
 18:     int i,j,rt,start,end,point;
 19:     b_init(); screen(1,1,1,1); b_csw(0);
 20:     mouse(4); mouse(1); msarea(0,0,119,63);
 21:     do{
 22:         cls(); color(7); rt=0;
 23:         for(i=0;i<7;i++){
 24:             a[i].moji='A'+i;
 25:             b[i]=&a[i];
 26:             d[0][i]=&a[i];
 27:         }
 28:         for(i=1;i<=MIX;i++){
 29:             randam(&start,&end,&point);
 30:             hantei(start,end,point);
 31:             sort();
 32:             for(j=0;j<7;j++) d[i][j]=b[j];
 33:         }
 34:         locate(0,1);  printf("* * * * * * * *");
 35:         locate(2,3);  printf("見"); fill(15,47,31,63,10);
 36:         locate(6,3);  printf("戻"); fill(47,47,63,63,10);
 37:         locate(12,3); printf("終"); fill(95,47,111,63,10);

 38:         while(rt>=0){
 39:             locate(0,0);
 40:             for(i=0;i<7;i++)
 41:                 printf(" %c",b[i]->moji);
 42:             for(i=0;i<7;i++)
 43:                 if(b[i]->moji!='A'+i) break;
 44:             if(i==7) break;
 45:             rt=input(&start,&point);
 46:             switch(rt){
 47:                 case -1:
 48:                 case  1: break;
 49:                     case  0: end=start+2;
 50:                      hantei(start,end,point);
 51:                      sort();
 52:             }
 53:         }
 54:         if(rt>=0){
 55:             locate(0,2);  printf("O.K!!");
 56:             locate(2,3);  printf("  ");fill(15,47,31,63,0);
 57:             locate(6,3);  printf("続");
 58:             locate(12,3); printf("終");
 59:             rt=input();
 60:         }
 61:     }while(rt!=-1);
 62:     b_exit(0);
 63: }
 64: void randam(int *start,int *end,int *point)
 65: {
 66:     *start=(int)(5*rnd());
 67:     *end=*start+2;
 68:     while(1){
 69:         *point=(int)(8*rnd());
 70:         if(*point<*start || *point>*end+1) break;
 71:     }
 72: }
 73: void hantei(int start,int end,int point)
 74: {
 75:     int i,count=0;
 76:     if(point<start){
 77:         for(i=0    ;i<point;i++) b[i]->no=count++;
 78:         for(i=start;i<=end ;i++) b[i]->no=count++;
 79:         for(i=point;i<start;i++) b[i]->no=count++;
 80:         for(i=end+1;i<7    ;i++) b[i]->no=count++;
 81:     }else{
 82:         for(i=0    ;i<start;i++) b[i]->no=count++;
 83:         for(i=end+1;i<point;i++) b[i]->no=count++;
 84:         for(i=start;i<end+1;i++) b[i]->no=count++;
 85:         for(i=point;i<7    ;i++) b[i]->no=count++;
 86:     }
 87: }
 88: void sort(void)
 89: {
 90:     int i,j,s,min;
 91:     for(i=0;i<6;i++){
 92:         min=b[i]->no; s=i;
 93:         for(j=i+1;j<7;j++){
 94:             if(min>b[j]->no){
 95:                 min=b[j]->no; s=j;
 96:             }
 97:         }c=b[i]; b[i]=b[s]; b[s]=c;
 98:     }
 99: }
100: int input(int *start,int *point)
101: {
102:     int i,j,flg,dx,px,py,br,bl,x,y;
103:     x=y=flg=0;
104:     while(1){
105:         msstat(&x,&y,&bl,&br); mspos(&x,&y);
106:         px=(int)x/16; py=(int)y/16;
107:         if(bl==-1 && flg==0 && py==0) {
108:             if(px>4) px=4;
109:             *start=px; dx=px; flg=1;
110:             fill(px*16+4,0,px*16+50,15,5);
111:           }
112:         if(bl==-1 && flg==1 && py==1) {
113:             *point=px; flg=0;
114:             fill(dx*16+4,0,dx*16+50,15,0);
115:             return(0);
116:         }
117:         if(br==-1 && flg==1){
118:             fill(dx*16+4,0,dx*16+50,15,0);
119:             flg=0;
120:         }
121:         if(bl==-1 && py==3){
122:             switch(px){
123:                 case 1: for(i=0;i<=MIX;i++){
124:                             locate(25,25-i);
125:                             for(j=0;j<7;j++)
126:                                 printf(" %c",d[i][j]->moji);
127:                         }break;
128:                 case 3:    fill(dx*16+4,0,dx*16+50,15,0);
129:                     for(i=0;i<7;i++) b[i]=d[MIX][i];
130:                     return(1);
131:                 case 6:    return(-1);
132:             }
133:         }
134:     }
135: }
```

リスト2　SIGMA.BAS

```
10 /*Tayler 展開
20 int n,s,i,j,k,l,fp,gt
30 dim float x(100),y(100),tx(100),ty(100)
40 dim float b(100,101),m(100),a(100)
50 float p,q,r
60 str dummy,fname : str kaku=".DAT",dir="¥DATA¥"
70 screen 2,0,1,1 :  width 96
80 locate 20,10,0 : print "Dataの入力方法は?"
90 locate 15,13,0 : print "1- ファイルを作らずに入力する。"
100 locate 15,15,0 : print "2- ファイルを作ってないので、Dataファイルを作
成する。"
110 locate 15,17,0 : input "3- もうファイルは作ってあるので、そちらから読む。
",l
120 switch l
130   case 1  : datainput() : break
140   case 2  : makefile()  : break
150   case 3  : readfile()  : break
160   default : print " "
170          print " 何を打ちこんどんじゃい! やり直し! "  : end
180 endswitch
190 sisuu()
200 cls : locate 20,10,0
```

▶表紙が怖いよ～。気の弱い赤ちゃんに見せたら絶対泣くぞ。　谷川　哲夫(30)埼玉県

```
210 print " 入力した Data を表示しますか ? "
220   locate 25,13,0 : print "1-はい "
230   locate 25,15,0 : print "2-いいえ "
240   locate 25,17,0 : input "3-初めからやり直し    ",j
250 switch j
260   case 1  :      dataprint()          : break
270   case 2  :                           : break
280   case 3  : print " RUN again .... "  : end
290   default :      dataprint()          : break
300 endswitch
310 cls
320 locate 22,10,0:print "これから計算します 宜しいですか "
330 locate 18,12,0:print "1--計算してよい   "
340 locate 18,14,0:input "2--入力値に間違いがあるのでもう一度...",k
350 switch k
360   case 2  : locate 40,20,0 : print "やり直し... " : end
370   default : break
380 endswitch
390 datatrans()
400 compute()
410 output()
420 cls
430 locate 30,10,0 : print "グラフはどうしますか? "
440 locate 30,12,0 : print "1--- 是非みてみたい "
450 locate 30,13,0 : input "2--- 見なくてよい       ",l
460 switch l
470   case 1  : graph() : break
480   default : break
490 endswitch
500 print "    X座標に対してYの値を求めることが出来ますが。"
510 print "                   1---求める。"
520 input "                   2---遠慮する       ",l
530 switch l
540   case 1  : trend() : break
550   default : break
560 endswitch
570 end
580 /* メインプログラム 終了
590 func  sisuu()
600   cls
610   locate 20,10,0
620   input "展開項の数を入力してください        ",s
630   locate 20,12,0
640   print "展開項の指数部分はどうしますか?       "
650   locate 13,14,0
660   print "1---最大値と最小値を指定して等間隔の累乗にする。"
670   locate 13,16,0
680   print "2---自分で任意に指定する      "
690   locate 20,18,0 : input " ",l
700   switch l
710           case 1  : jun() : break
720           case 2  : nin() : break
730           default : nin() : break
740   endswitch
750 endfunc
760 func  jun()
770   float mmax,mmin
780   locate 15,20,0
790   input "指数部分の最大値を入力して下さい    ",mmax
800   locate 15,22,0
810   input "指数部分の最小値を入力して下さい    ",mmin
820   for i=0 to s-1
830           m(i+1)=(mmax-mmin)/(s-1)*i+mmin
840   next
850 endfunc
860 func  nin()
870   cls:locate 33,7,0 : print " 指数部分を入力して下さい"
880   for i=1 to s
890           l=i+9
900           locate 30,1,1
910           print "m(";i;")= " : locate 38,1,1 : input "",r
920           m(i)=r
930   next
940 endfunc
950 func  dataprint()
960   cls : locate 30,4,0 : print "Dataを出力します    "
970   for i=1 to n
980           l=i+5
990           locate 20,1,0
1000          print "x(";i;"),y(";i;")= ";x(i);" , ";y(i)
1010  next
1020  print "何かkeyを押して下さい"
1030  dummy=inkey$
1040  cls : locate 30,4,0 : print "Dataを出力します    "
1050  for i=1 to s
1060          l=i+5
1070          locate 20,1,0
1080          print "m(";i;")= ";m(i)
1090  next
1100  print "何かkeyを押して下さい"
1110  dummy=inkey$
1120  cls
1130 endfunc
1140 func  datatrans()
1150  cls : locate 20,12,0
1160  print "グラフはどのグラフですか? "
1170  locate 30,15,0:print " 1--- X~Y グラフ "
1180  locate 30,16,0:print " 2--- Semi-log グラフ"
1190  locate 30,17,0:input " 3--- log-log グラフ  ",gt
1200  locate 17,20,0:print "ただ今計算中です。しばらくお待ち下さい 。"
1210  switch gt
1220          case 1  : for i=1 to n
1230                          tx(i)=x(i)
1240                          ty(i)=y(i)
1250                    next
1260                    break
1270          case 2  : for i=1 to n
1280                          tx(i)=x(i)
1290                          ty(i)=log(y(i))
1300                    next
1310                    break
1320          case 3  : for i=1 to n
1330                          tx(i)=log(x(i))
1340                          ty(i)=log(y(i))
1350                    next
1360                    break
1370          default : for i=1 to n
1380                          tx(i)=x(i)
1390                          ty(i)=y(i)
1400                    next
1410                    break
1420  endswitch
1430 endfunc
1440 func  compute()
1450  for i=1 to s
1460  for j=1 to s
1470  for k=1 to n
1480          b(i,j)=b(i,j)+pow(tx(k),m(i))*pow(tx(k),m(j))
1490  next
1500  next
1510  next
1520  for i=1 to s
1530  for k=1 to n
1540          b(i,s+1)=b(i,s+1)+ty(k)*pow(tx(k),m(i))
1550  next
1560  next
1570  /* ここから掃き出し法
1580  for k=1 to s
1590    for i=k+1 to s+1
1600          b(k,i)=b(k,i)/b(k,k)
1610    next
1620    b(k,k)=1
1630    for i=1 to k-1
1640      for j=k+1 to s+1
1650          b(i,j)=b(i,j)-b(i,k)*b(k,j)
1660      next : b(i,k)=0
1670    next
1680    for i=k+1 to s
1690      for j=k+1 to s+1
1700          b(i,j)=b(i,j)-b(i,k)*b(k,j)
1710      next : b(i,k)=0
1720    next
1730  next
1740 endfunc
1750 func  output()
1760  cls : locate 20,12,0 : print "こんなん出ましたけど...."
1770  locate 15,15,0
1780  for i=1 to s
1790          a(i)=b(i,s+1)
1800  next
1810  for i=1 to s
1820          print a(i);"*X^";m(i);" + ";
1830  next
1840  print "........"
1850  print "        何か Key を押して下さい    "
1860  dummy=inkey$
1870 endfunc
1880 func  datainput()
1890  cls : locate 20,5,1
1900  input "Data の数を入力して下さい ",n
1910  locate 20,7,1 : print "Data を入力して下さい  "
1920  for i=1 to n
1930    l=i+8
1940    locate 25,1,0
1950    input "(x,y)= ",p,q
1960    x(i)=p : y(i)=q
1970  next
1980  print "     Data の入力は終わりました 何か Key を押して下さい"
1990    dummy=inkey$
2000 endfunc
2010 func  graph()
2020  cls : screen 2,0,1,1  : int k2,l2,cc1,cc2,cc3,cc4
2030  float px,py,px2,py2,maxtx,mintx
2040  float maxty,minty,abx,dcx,ady,bcy
2050  maxtx=tx(1) : mintx=tx(1)
2060  maxty=ty(1) : minty=ty(1)
2070  for i=1 to n
2080          if tx(i)>maxtx then   maxtx=tx(i)
2090          if tx(i)<mintx then   mintx=tx(i)
2100          if ty(i)>maxty then   maxty=ty(i)
2110          if ty(i)<minty then   minty=ty(i)
2120  next
2130  /*
2140  /*   a          d
2150  /*     |---------|
2160  /*     |         |
2170  /*     |         |
2180  /*     |_________|
2190  /*   b          c
2200  /*
2210  dcx=maxtx+(maxtx-mintx)*0.1#
2220  abx=mintx-(maxtx-mintx)*0.1#
2230  ady=maxty+(maxty-minty)*0.1#
2240  bcy=minty-(maxty-minty)*0.1#
2250  /* グラフを書く
2260  cc1=rgb(21,21,21)
2270  palet(0,cc1)
2280  cc2=rgb(0,0,0)
2290  palet(1,cc2)
2300  box(88,62,688,462,1)
2310  /* 実験値を記入
2320  cc3=rgb(10,3,0)
2330  palet(2,cc3)
2340  for i=1 to n
2350          px=88+(tx(i)-abx)*600/(dcx-abx)
2360          py=462-(ty(i)-bcy)*400/(ady-bcy)
2370          k=int(px)
2380          l=int(py)
2390          circle(k,1,5,2,0,360,256)
2400  next
2410  /* 近似式のグラフを書く
2420  cc4=rgb(3,10,0)
2430  palet(3,cc4)
2440          q=(dcx-abx)/600+abx
```

▶「GENOCIDE」って大量殺戮という意味だったんですね（模試の注釈に載ってた）。
松居　啓樹(18)富山県

```
2450            p=0
2460            for j=1 to s
2470                    if q<0 and m(j)<>int(m(j)) then {break}
2480                    p=p+a(j)*pow(q,m(j))
2490            next
2500            p=462-(p-bcy)*400/(ady-bcy)
2510            k2=88
2520            l2=int(p)
2530    for i=1 to 300
2540                    q=2*i*(dcx-abx)/600+abx
2550                    p=0
2560            for j=1 to s
2570                    if q<0 and m(j)<>int(m(j)) then {break}
2580                    p=p+a(j)*pow(q,m(j))
2590            next
2600                    p=462-(p-bcy)*400/(ady-bcy)
2610            k=2*i+88
2620            l=int(p)
2630            if l<462 and l>62 then   line(k,l,k2,l2,3)
2640            k2=k : l2=l
2650    next
2660    print "こんなもんでどうでっしゃろ? "
2670    print "なんか きい を 押してくれまへんか? "
2680    dummy=inkey$
2690    output()
2700 endfunc
2710 func  readfile()
2720    dim float data(1)
2730    cls : locate 10,10,1
2740    linput "ファイルの名前を入力してください ";fname
2750    fname=dir+fname+kaku
2760    fp=fopen(fname,"r")
2770    fread(data,1,fp)
2780    n=int(data(0))
2790    k=n+1
2800    fseek(fp,0,0)  : fread(x,k,fp)
2810    fseek(fp,-8,1) : fread(y,k,fp)
2820    fclose(fp)
2830    locate 2,12,0
2840    print "DATAの読み取りが終了しました。何かKEYを押して下さい  "
2850    dummy=inkey$
2860 endfunc
2870 func  makefile()
2880    dim float fw(201)
2890    cls : locate 10,10,1
2900    print "Bドライブにデータファイルを作ります。"
2910    locate 10,13,0 : print "準備はよろしいですか?  "
2920    locate 10,15,0 : print "よろしければ何かKEYを押して下さい。"
2930    dummy=inkey$  : cls
2940    repeat
2950        locate 15,12,1
2960        linput "ファイル名を入力して下さい。";fname
2970        locate 5,14,0
2980        print fname;"というDATAファイルを作成します。"
2990        locate 10,16,0 : print "よろしいですか? "
3000        locate 15,18,0 : print "1----Yes"
3010        locate 15,19,0 : input "2----No  ",l
3020        cls
3030    until l=1
3040    datainput()
3050    fname=dir+fname+kaku
3060    fw(0)=n
3070    for i=1 to n
3080            fw(i)=x(i)
3090            fw(i+n)=y(i)
3100    next
3110    k=2*n+1
3120    fp=fopen(fname,"c")
3130            fwrite(fw,k,fp)
3140    fclose(fp)
3150 endfunc
3160 func trend()
3170    print "指定されたX座標に対して推測されるY座標を求められます。"
3180    repeat
3190            print " 知りたいX座標を入力して下さい。"
3200            input "         X=    ",q   :   p=0
3210            if gt=3  then q=log(q)
3220            for i=1 to s
3230                    p=p+a(i)*pow(q,m(i))
3240            next
3250            switch gt
3260                    case 2   : p=exp(p) : break
3270                    case 3   : p=exp(p) : break
3280                    default : break
3290            endswitch
3300            print "    Y=  ",p
3310            print "終了したい時は 0 を入力して下さい。"
3320            input "もう1回計算したい時はそのまま RETURN して下さい。",l
3330    until l=0
3340 endfunc
```

リスト3 SP_ALL.S

```
  1: *-----------------------------------------------------------
  2: *SP_ALL.S(.X)                          First   '92 8
  3: *   ｽﾌﾟﾗｲﾄをすべて同時に表示           Remake  '93 9   OWL.AKI
  4: *___________________________________________________________
  5:
  6:    .include        DOSCALL.MAC
  7:    .include        IOCSCALL.MAC
  8:
  9: SET_RAS    macro   nextr,rasno         *割り込みﾙｰﾁﾝ共通部分ﾏｸﾛ
 10:    move.l  #nextr,$000138
 11:    move.w  #rasno,$e80012
 12:    movem.l (sp)+,a0-a1
 13:    ori.b   #$40,$e88007
 14:    rte
 15: endm
 16:
 17: .text
 18:
 19: main:
 20:    suba.l  a1,a1
 21:    IOCS    _B_SUPER
 22:    move.l  d0,super
 23:    moveq   #$ff,d1
 24:    IOCS    _CRTMOD
 25:    move.w  d0,crtmod
 26:    moveq   #6,d1                 *(512) 6→4
 27:    IOCS    _CRTMOD               *^512*512ﾄﾞｯﾄﾓｰﾄﾞ時変更箇所
 28:    IOCS    _OS_CUROF
 29:    IOCS    _SP_ON
 30:
 31:    bsr     SET_SP                *初期化ﾙｰﾁﾝｺｰﾙ
 32:    bsr     INT_START             *初回割り込みｽﾀｰﾄ
 33:
 34: *終了待ちループ
 35:    moveq   #1,d7                 *Palet No.
 36: loopkey:
 37:    moveq   #0,d1                 *ｷｰｺｰﾄﾞｸﾞﾙｰﾌﾟ0でEXIT
 38:    IOCS    _BITSNS
 39:    tst.b   d0
 40:    bne     exit
 41:    moveq   #6,d1                 *ｷｰｺｰﾄﾞｸﾞﾙｰﾌﾟ6で
 42:    IOCS    _BITSNS               *色変更
 43:    tst.b   d0
 44:    beq     loopkey
 45: keyoff:
 46:    moveq   #6,d1                 *ｷｰ離し待ち
 47:    IOCS    _BITSNS
 48:    tst.b   d0
 49:    bne     keyoff
 50:
 51: COL_CHANGE:                               *ｽﾌﾟﾗｲﾄﾊﾟﾚｯﾄ+1(0～15)
 52:    lea.l   SP_y_buf1+2(pc),a0
 53:    .dcb.l  256,$10875888
 54:                    *move.b d7,(a0)/addq.l #4,a0 が 256個
 55:    addq.w  #1,d7
 56:    cmpi.w  #16,d7
 57:    bne     loopkey
 58:    moveq   #0,d7
 59:    bra     loopkey
 60:
 61: exit:
 62:    andi.b  #$bf,$e88007          *割り込み禁止を忘れずに
 63:    movea.l super(pc),a1
 64:    IOCS    _B_SUPER
 65:    move.w  crtmod(pc),d1
 66:    IOCS    _CRTMOD
 67:    IOCS    _OS_CURON
 68: keybufclr:
 69:    IOCS    _B_KEYSNS
 70:    tst.l   d0
 71:    beq     end
 72:    IOCS    _B_KEYINP
 73:    bra     keybufclr
 74: end:
 75:    DOS     _EXIT
 76:
 77: *ワーク
 78: super:              .dc.l   0
 79: crtmod:             .dc.w   0
 80:
 81: *↓ここから最後までを組み込み用にｺﾋﾟｰして使えます。
 82: *先頭のﾏｸﾛもｺﾋﾟｰすること。色変更は上のﾙｰﾁﾝを参照のこと。
 83: *-----------------------------------------------------------
 84: *割り込みスタート
 85: INT_START:                            *割り込み奨励ｷｬﾝﾍﾟｰﾝのｺｰﾅｰ↓
 86:    andi.b  #$bf,$e88007          *割り込み禁止(割り込みの基本)
 87:    ori.b   #$40,$e88013          *ﾏｽｸははがす
 88:    move.l  #spint1,$000138 *ﾍﾞｸﾀをｾｯﾄして..
 89:    move.w  #423,$e80012          *(512) 423->231->ﾗｽﾀｰ指定
 90:    ori.b   #$40,$e88007          *割り込み許可(ｽﾀｰﾄ)
 91:    rts                           *これが基本ﾊﾟﾀｰﾝ
 92:
 93: *-----------------------------------------------------------
 94: *割り込み処理本体
 95: spint1:                 *画面3/4の位置のﾗｽﾀｰの処理
 96:    andi.b  #$bf,$e88007
 97:    movem.l a0-a1,-(sp)
 98:    movea.l #$eb0002,a0         *ｽﾌﾟﾗｲﾄﾊﾟﾀｰﾝ0～63を
 99:    lea.l   SP_y_buf1(pc),a1*0～63にｾｯﾄ
100:    .dcb.l  64,$20d95888
101:                    *move.l (a1)+,(a0)+/moveq.l #4,a0 が 64個
102:    SET_RAS spint2,553            *(512) 553→297
103:
104: spint2:                 *画面最下位置のﾗｽﾀｰの処理
105:    andi.b  #$bf,$e88007
106:    movem.l a0-a1,-(sp)
107:    movea.l #$eb0202,a0           *ﾊﾟﾀｰﾝ64～127を64～127にｾｯﾄ
108:    lea.l   SP_y_buf2(pc),a1
109:    andi.w  #$ff,$eb0808          *ｽﾌﾟﾗｲﾄを消して時間を稼ぐ
110:    .dcb.l  64,$20d95888          *spint1と同じ
111:    ori.w   #$200,$eb0808
112:    SET_RAS spint3,167            *(512) 167→103
113:
114: spint3:                 *画面1/4の位置のﾗｽﾀｰの処理
115:    andi.b  #$bf,$e88007
116:    movem.l a0-a1,-(sp)
117:    movea.l #$eb0002,a0           *ﾊﾟﾀｰﾝ128～191を0～63にｾｯﾄ
118:    lea.l   SP_y_buf3(pc),a1
119:    .dcb.l  64,$20d95888          *spint1と同じ
120:    SET_RAS spint4,295            *(512) 295→167
```

```
121:
122: spint4:                    *画面1/2の位置のﾗｽﾀｰの処理
123:     andi.b   #$bf,$e88007
124:     movem.l  a0-a1,-(sp)
125:     movea.l  #$eb0202,a0      *ﾊﾟﾀｰﾝ192～256を64～127にｾｯﾄ
126:     lea.l    SP_y_buf4(pc),a1
127:     .dcb.l   64,$20d95888     *spint1と同じ
128:     SET_RAS  spint1,423       *(512) 423→231
129:
130: *-----------------------------------------------------------
131: *初期化ﾙｰﾁﾝ
132: SET_SP:                          *ｽﾌﾟﾗｲﾄｽｸﾛｰﾙﾚｼﾞｽﾀに
133:     moveq    #0,d0               *あらかじめ、X座標を設定
134:     movea.l  #$eb0000,a0         *同時にｽﾌﾟﾗｲﾄ表示
135: loop_spon:                       *Y座標、反転ﾋﾞｯﾄはｸﾘｱ,ﾊﾟﾚｯﾄ0
136:     move.w   d0,d1
137:     andi.w   #$000f,d1
138:     lsl.w    #4,d1
139:     addi.w   #$10,d1
140:     move.w   d1,(a0)+
141:     move.l   #0,(a0)+
142:     move.w   #$0003,(a0)+
143:     addq.w   #1,d0
144:     cmpi.w   #128,d0
145:     bne      loop_spon
146:
147: MAKE_buf:   *ｽﾌﾟﾗｲﾄのY座標をあらかじめ計算しておく
148:     moveq    #0,d0
149:     lea.l    SP_y_buf1(pc),a0
150: loop_mb:
151:     move.w   d0,d1
152:     andi.w   #$fff0,d1          *ﾙｰﾌﾟ内を最適化したいよぅ
153:     addi.w   #$10,d1
154:     move.w   d1,(a0)+
155:     move.w   d0,(a0)+
156:     addq.w   #1,d0
157:     cmpi.w   #256,d0
158:     bne      loop_mb
159:     rts
160:
161: *-----------------------------------------------------------
162: .bss
163:
164: SP_y_buf1: .ds.w   64*2
165: SP_y_buf2: .ds.w   64*2
166: SP_y_buf3: .ds.w   64*2
167: SP_y_buf4: .ds.w   64*2
168:
169: .end
```

ぷろぐらむ風まかせ 1

どもどもどもーっ。いままで「ショートプロぱーてぃ」を読んでいてくれた皆さんは知っていると思いますけど，この「ぷろぐらむ風まかせ」の前には「ぱーてぃハンズ」っていうプログラムを少しずつ作っていくコーナーがあったんです。ま，この「ぷろぐらむ風まかせ」も同じようにプログラムを作っていくんですが，このコーナーではプログラムに関するいろいろ細かい部分にもふれながらやっていこうと思っています。さ，プログラムの道を歩きましょう！

動かないよと思う前に

さて，さっそく今月のプログラム……に入る前にちょっと寄り道します。今月の「ショートプロぱーてぃ」なんですが，3本目のプログラムにSP_ALL.Sというプログラムが掲載されていますね。

このSP_ALL.Sは256個のスプライトを表示するためにラスター割り込みを使っています。

割り込み関係のプログラムっていうのは，プログラムが間違っていたりすると，原因不明のエラーが出たり，いきなり暴走したりすることが多いんですよね。それというのも，X68000で使っている68000CPUっていうのは，普段はユーザーモードっていうモードで動いているんです。このユーザーモードでは68000CPUはプログラムが変な動きをしないように見張ってくれています。なにかあってもバスエラーやアドレスエラーというエラーを起こして止まり，OSの制御に戻すようにしてくれているのです。ところが，割り込み関係の制御ではスーパバイザモードというモードになるんです。そのモードでは，たとえばOSでしか使ってはいけない領域をいじったり，I/Oを直接操作したり，ということができるのです。ところがCPUはなにが起こっても注意してくれません。そのために割り込み関係のプログラムで作り方を間違えると普通のプログラムでは見たことのないような変な動きをすることがあるんですね。

したがってこのSP_ALLのプログラムを実行して変になったときには，リスト中の注釈で「割り込み～」になっている箇所をもう一度見直してみてください。割り込みがぶつかるような常駐ソフトが原因のときもありますけど。

でもってプログラムを作る……

さてさて，それではプログラムの話を。

今回のテーマは「なんてったって入力～！」なのであります。

入力装置。あるいは入力デバイス。ま，ひらたくいえば，「コンピュータについている入力するためのもの」。なんか説明になってないよーな気もしますがそういうものです。で，もしも入力装置がなかったら，この世にコンピュータが存在する意味が全然なくなってしまいますよね。そういう意味で私はキーボードを作った偉大な先達（誰だか知らないけど）に感謝しているのであります。

で，X68000にはいろんな入力装置があるんです。まずはキーボード。でもってジョイスティック。マウスなんてのもそうですし，あとうらやましいところではイメージスキャナなんかも……。ああ，私もお金があれば……。みんなビンボーが悪いんや～。ま，それはともかく，この入力デバイスたちってのはどうやって使うんでしょうか？

マウスを使う

ゲームなどをするときにいろいろな入力デバイスが使えますが，なんといってもX68000を持っていれば，誰でも持ってるのがマウスなんですよね。

マウスを使う例というとまっさきに思いつくのはSX-WINDOWなどのウィンドウシステムでしょう。マウスカーソルをマウスで上下左右に操って，ボタンを押してコマンド実行。

しかし，マウスってーのはキーボードやジョイスティックとはちょっとばかり入力情報の質が違います。それはマウスの底にあるボール，これがどのくらい回ったかをみるものなのです。ジョイスティックのように，どの方向に入力したかっていうのが決められないんです。つまり，動かしたかどうかではなく，「どのくらい動いたか」っていうのが返ってくるんですね。

さて，このマウスカーソルの動きやボタンが押されたかどうかは，X-BASICではマウスを使う命令はMSで始まる命令で制御できます。

MSAREA()
MSSTAT()
MSBTN()
MSPOS()

という4つの命令です。ゲームで使いやすいのはMSSTAT()とMSPOS()あたりでしょうか。

MSSTAT()はマウスの状態を調べる命令で，

MSSTAT(x,y,br,bl)

としてやるとx，yにマウスのx方向y方向の移動量が入ります。bl，brは左右のボタンの状態を示す値が返ってきます。

MSPOS()はマウスカーソルの座標を返すので，

MSPOS(x,y)

とするとx，yにマウスカーソルのx，y座標が返ってきます。

MSSTAT()とMSPOS()，それぞれのx，yが，どのように違うかというと，MSSTAT()のほうは「移動量」なんです。たとえば，マウスをすっと動かしたときにはx＝125，y＝86で，止めるとx＝y＝0が返ってきます。MSPOS()は「座標」ですから，動かしてx＝125，y＝86になったら，止めてもx＝125，y＝86のままです。

ところで，MSSTAT()ではマウスカーソルの座標がわからないし，MSPOS()ではマウスのボタンがどうなっているかわからないですよね。それではいまのマウスカーソルの座標を知り，なおかつマウスのボタンの状態を知るにはどうしたらいいんでしょう。そうなんです。まず，

MSPOS(x,y)

で座標を知ってから，

MSSTAT(dummy,dummy,br,bl)

でボタンの状態を見なくちゃいけないんですよ。めんどくさいですね。さらにいうとマウスの場合はX-BASICでは，初めからマウスが使える状態にあるわけではないのでプログラムを始める前に初期化処理が必要ですから，

MOUSE(0)

などとして使えるようにしてやらなくちゃいけないんですね。

ではちょいとこいつを使って実際にプログラムを組んでみようか……と思ったけど，ここまでにしましょう。来月はほかの入力装置を見てからプログラムを作ってみようと思います。

CによるデバイスドライバのP開発実験 PART 2.

電机本舗 由井 清人 Yui Kiyoto

C言語でデバイスドライバを記述するための実験編その2です。今回はMS-DOSやHuman68kのデバイスドライバの基本構造と内部処理についてまとめてました。次回よりいよいよX68000通信ドライバの制作編の予定です。

前回に引き続きC言語によるブロック型デバイスドライバ開発をレポートします。

前回はアセンブラで記述されたSRAMディスクドライバをもとに機械的にC言語のソースを作成しました(人間ディスコンパイル?)。これは，すでに動いているプログラムを変造することにより，開発のノウハウを上手に吸い上げるためです。

これにより，本来アセンブラで組むべきであったデバイスドライバをどうにかC言語で作れるようになりました。

前回は，C言語でのデバイスドライバの作成実験であったといえるでしょう。ただし，問題点として，C言語へと機械的に移植したため，SRAMディスクドライバのアルゴリズムはわかりにくいものになったという点があります。

今回は，デバイスドライバプログラムのアルゴリズムに重点を置き，ドライバ内部ではどのようにしてディスク装置を処理をしているかという点に的を絞ってレポートします。

まず，初めに今回作ったプログラムを紹介します。これは，前回のSRAMディスクドライバをアルゴリズムに従い書き直したものです。内訳を以下に示します。

リスト1　デバイスドライバメインプログラム(アセンブラ)
リスト2　C言語による処理本体
リスト3　ダミープログラム（コラム参照)

ディスクデバイスに関する考察

さて，ディスクドライバのアルゴリズムを考える前に，ディスク装置そのものの構造を考えてみましょう。ブロック型デバイスはディスク装置を制御するためのものですから，FDD（フロッピーディスクドライブ）ないしHDD（ハードディスクドライブ）の構造をよく理解する必要があります。

ディスク装置といってもたくさんの種類があります。とりあえず，ここではFDDを例に構造を説明していきます。

いうまでもなくFD（フロッピーディスク）は円盤状になっています。そしてバームクーヘン状に区画を仕切って記憶領域を確保管理しています。

これを示すと図1になります。

FDは円盤の裏と表にそれぞれトラックと呼ばれる同心円状の記憶領域を持っています。そして，トラックは内部にセクタと呼ばれる小区画を持っています。ですからFDの記憶量は次の式で算出できます。

記憶容量＝トラック数×セクタ数×1セクタのバイト数×面数

ちなみに1セクタはフォーマット形式により異なりますが一般的には128バイトから2048バイトの容量を持ちます。Human68k(FD)では1024バイト/セクタが採用さ

ダミーファイルD2.S

今回，D2.Sというファイルを作成しました。これは，なんのことはなく，ただ　inittblendというラベルを持っているだけです。

なぜ，このような一見無意味なファイルをつけたかというと，このファイルをプログラムの最後に結合することによりプログラムの尻を把握したかったからです。

デバイスドライバはシステム起動時にメモリに読み込まれます。このときに，デバイスドライバの後ろに，またなにかほかのプログラムなどが読み込まれます。ですから，正しくプログラムの最後を教えてあげないと障害が考えられます。

アセンブラですと最終アドレスの算出はプログラムの最後にラベルをつければ，このラベルが最終アドレスとして扱えるようになり簡単です。

しかし，C言語はラベルの概念がありませんし（強いていえば誰も使わないgoto文がラベルを使いますね），変数や関数の位置するアドレスはコンパイラが自動的に生成します。ですから，このようなときにはなはだ不便です。というわけで，ひとつの手段として，このようなダミーファイルを使ってみました。

リンクするときに，結合する最後のオブジェクトファイルにこのダミーファイルを指定すれば，順当に考えてプログラムの最後に結合されるはずです（あくまで，可能性の問題)。ですから，このようにして，Human68kに渡す最終アドレスを得ているのです。

れています。

さて，このような構造をとっていますが，トラック数，セクタ，円盤の表裏でリード/ライトを指定するのは面倒です。そこで，一般的なディスク管理システム（DOS）では円盤の表裏，全トラック上に存在するセクタに一連の番号を振り，この番号でセクタを指定します。ですから，プログラムはこの論理番号を管理すればよいことになります。

これを論理セクタ番号といいます。論理セクタを示すと表1のようになります。こうなると記憶単位が複数バイトであることを除けば半導体メモリとさして違いがないことがわかります。

RAMディスクは，本来半導体メモリですから，このような構成をとる必要はありません。ですが，ディスク装置との互換性をとるために構成を模倣しています。

ディスクのデータ構造

ディスクを上手に利用するために，MS-DOS（Human68kのディスク制御はMS-DOS互換）は次のように領域を分けて使用しています（図2）。

ディスクの各領域の内訳は次のようになります。

1) IPL(Initial Program Loader)領域
2) FAT(File Allocation Table)領域
3) ルートディレクトリ領域
4) データ領域

それぞれについて解説しましょう。

1) IPL領域

IPLというのはパソコンの起動プログラムのことで，IPL領域は通常はディスクの先頭に置かれており，起動プログラムそのもの，もしくは起動プログラムの格納場所が入っています。FDよりX68000を起動する場合，X68000はまずこのデータを読み取り，システムプログラムを実行します。

このようなわけでSRAMディスクではIPL領域は予約領域で確保しておけばよいことになります。しかし実際には，IPL領域とはいうものの，ここにはディスク構造を定義する情報が入っていますので，注意する必要があります。IPL領域の内訳を表2に示します。

表のうちBPB(BIOS Parameter Block)はディスクサイズ，セクタのバイト数などを定義しており重要です。

BPBの詳細に関してはコラムを参照してください。このあと，ディスク管理用語としてFAT，クラスタというものが出てくるので，とりあえずざっと全体像を理解して，詳細はあとからコラムより理解するとよいでしょう。

2) FAT (File Allocation Table)領域

これはファイルがディスク中のどこに存在するかを管理するものです。ディレクトリ情報の本体という性質のものです。

DOSはディスク管理情報として，現在使用されているセクタと，そうでないセクタをこのFATという表にして持っています。

図1　フロッピーディスクの構造

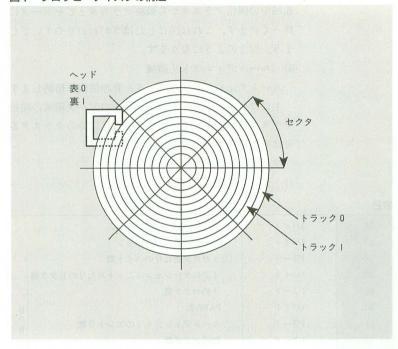
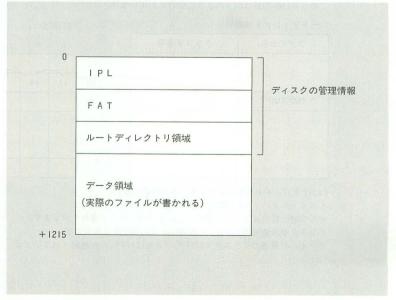



表1　論理セクタの並び方

ヘッド	0（表）	1（裏）
トラック ＼ セクタ	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
0	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15
1	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
2	32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47
:	:	:
76	································	························ 1215

論理セクタ番号

図2　論理セクタ空間の構成

まず，FATはクラスタという単位でセクタを管理しています。セクタ単位ですと細かすぎて扱いにくいので，数セクタ単位で管理しています。

FATはチェイン構造をとっており，ディスクのデータ領域中の開始クラスタから最終クラスタまでをデータに持っています。これは図にしたほうがわかりやすいでしょう。図3のようになります。

3)　ルートディレクトリ領域

ルートディレクトリのファイル管理情報を格納します。

主にファイル名と，そのファイルのデータ領域の開始クラスタ番号を持っています。2番目以降のクラスタ番号は先のFAT領域に格納されています。

ファイル管理情報は表3になります。

4)　データ領域

ここには，実際のファイルデータが格納されます。ただし，例外的にサブディレクトリは，ファイルと同じ扱いをうけるので，サブディレクトリの管理情報もここに格納されます。

デバイスドライバの実際

これまでのことより，ブロック型のデバイスドライバはIPL，FAT，ルートディレクトリ，データ領域を制御すればよいことになります。

デバイスドライバがOSにどのように組み込まれるか，また，どのようにしてデータの受け渡しをするかは前回に報告したとおりです。

今回は，デバイスドライバの内部に立ち入って，内部の処理プログラムを説明しながらIPLなどの領域との関係をここで解説します。

1)　SRAMディスクの構成

SRAMディスクドライバはSRAMのメモリをディスクにみたてRAMディスクを構築するものです。

SRAMの構成を図4に示します。メインメモリ上のED0000$_{\mathrm{H}}$～ED3FFF$_{\mathrm{H}}$に存在します。このうちED4000$_{\mathrm{H}}$～ED3FFF$_{\mathrm{H}}$の約15Kバイトの領域がRAMディスク用に用意されています。この空間をディスクの論理空間にみたて，セクタ単位で読み書きをし，疑似ディスクを構築します。

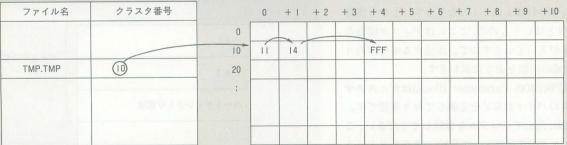
表4にSRAMディスクに関連する重要な項目を抜粋しておきます。

表4のうち，E8E00D$_{\mathrm{H}}$番地は特に重要です。この番地

表2

00	3バイト	IPLへのジャンプアドレス	
03	8バイト	メーカー名，バージョン	
0B	1ワード	1セクタあたりのバイト数	BPB
0D	1バイト	1アロケーションユニットあたりのセクタ数	BPB
0E	1ワード	予約セクタ数	BPB
10	1バイト	FATの数	BPB
11	1ワード	ルートディレクトリのエントリ数	BPB
13	1ワード	論理セクタ数	BPB
15	1バイト	メディアディスクリプタバイト	BPB
16	1ワード	1FATあたりのセクタ数	BPB
18	1ワード	1トラックあたりのセクタ数	
1A	1ワード	ヘッド数	
1C	1ワード	隠れたセクタ数	
1E～		IPL	

図3

いまルートディレクトリにあるTMP.TMPというファイルが
ディスク中の10，11，14番の3つのクラスタに書き込まれていたとします。
このときTMP.TMPは以下のように管理されています。

FATは3バイトもしくは4バイトのセルの表です。
3バイトか4バイトかはFD，HDの容量により異なります。
セル領域の数値はファイルで使っている次のクラスタ番号を示します。
クラスタが未使用のときはFAT領域のセル内容が0になります。
ファイルの最後のクラスタではFFF$_{\mathrm{H}}$（またはFFFF$_{\mathrm{H}}$）が格納されています。

の値によってSRAMの読み書きの禁止/解除をしています。

SRAMへ書き込みを行いたいときは，ここに31_Hを，禁止したいときは00_Hを設定しておきます。

デバイスドライバの全体構造

デバイスドライバはエントリルーチン（リクエストルーチン，割り込みルーチン）を介して，OSより呼ばれます。

エントリルーチンはentry（入口）ということからわかるようにHuman68kからの制御の入口です。すべて，ここを経由して制御のやりとりを行います。

この部分がデバイスドライバのプログラム本体です。図5にこの様子を示します。

OSはドライバに対応して，表3の命令を持っています。ですからドライバは通常はそれぞれに対応した内部処理プログラムを持っていると思ってください。

次に各内部プログラムの具体的な説明をしていきます。リスト2の中にそれぞれの関数が入っていますので参照してください。

1)　命令No.2，3，6，7，10，11，12

未使用命令，エラー命令。関数notcom()により対応。

BPBテーブル

BPB（BIOSパラメータブロック）はIPL領域の中に入っていますが，IPL(Initial Program Loader-初期化プログラム読み込み機構？）とは異なり，ディスクそのものを定義するものです。すなわち，論理セクタの個数，セクタのバイト数などです。これはBPBの内容を見れば明解です。ブロックデバイスは何MBのHDDであれ，ここを見ればデバイスドライバは同じように扱えるようになるわけです。

さて，今回のSRAMディスクでは次のようになります。

1セクタ　＝　1024バイト
1クラスタ　＝　1セクタ
FAT個数　＝　1
ルートディレクトリのエントリ数　＝　32
全領域のセクタ数　＝　16
メディアバイト　＝　19_H
FAT領域に使用するセクタ数　＝　1

余談ですがBIOSパラメータブロックのBIOS(Basic Input Output System)というのは，10年前に一世を風靡したインテル系8ビット用オペレーティングシステムCP/Mから使われだした用語です。8ビットマシン，16ビットマシン，CPUメーカーの違いはありますが，CP/M→MS-DOS→Human68kと互換性を維持しながら変化派生しているので，ひょっこり「言葉」が残るようです。

同様にFATというのは，10年以上前，まだ8ビット全盛のころの標準であったマイクロソフトBASICのディスク管理テーブルに由来しており，同様の管理方式であったと記憶しています。当時雑誌の解説特集の中ではFATという名称（もしくはこの方式？）はマイクロソフト固有のもので，ほかのコンピュータでは異なっており通じないと記述されていたように思います。なにはともあれ，技術は継承され改良されるのだと感心する次第です。

これらは使われていない命令です。ですから，これが呼ばれるとエラーコードをリクエストヘッダに設定してHuman68kに戻ります。

2)　命令No.0，初期化

デバイスドライバの初期化命令です。関数dskini()により処理を行っています。

リクエストヘッダへ各種パラメータをセットします（表5）。

また，SRAMを新しくRAMディスクとして使う場合にはフォーマットをかける必要があります。このときは，

表3

命令コード	要　求
0	初期化
1	ディスク交換チェック
2	未使用
3	IOCTRLによる入力
4	入力
5	ドライブコントロールシーケンス
6	エラー
7	エラー
8	出力（VERIFY OFF時）
9	出力（VERIFY ON時）
10	エラー
11	未使用
12	IOCTRLによる出力

図4　メモリ構成

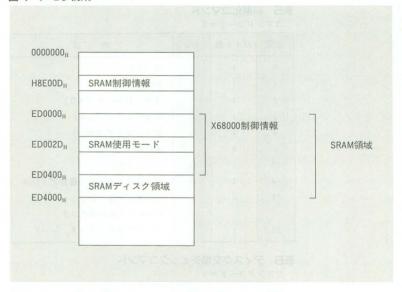



表4

$E8E00D_H$	SRAM制御情報 0　＝SRAM書き込み禁止 31_H　＝SRAM書き込み許可
$ED002D_H$	SRAM使用モード。$ED0100_H$～$ED3FFF_H$までの使い方を指定 0　＝使用しない 1　＝SRAMDISKとして使用 2　＝プログラムメモリとして使用

次の処理をします。

IPLセット

FATクリア

ルートディレクトリクリア

また，本プログラムは起動時にシフトキーを押していれば強制的にフォーマットをかけSRAMをRAMディスクとして強制初期化します。

3) 命令No.1，メディアの交換

ディスク交換のチェックを行います。関数mediac()により処理（表6）。

この命令はFD，MOのようなデバイスのための処理です。SRAMディスクでは関係ない命令ですが，SRAMが壊れていないか簡単にテストして結果をリクエストヘッダのデータバッファに格納しています。

4) 命令No.4，8，9，入出力処理（表7）

ディスクへの読み書き処理をする命令です。

これらは同じリクエストヘッダを持ちます。

各処理は，

① 入力処理，関数dskinp()

② 出力処理（書き込みチェックなし），関数dskout()

③ 出力処理（書き込みチェックあり），関数dskotv()

入出力データは，セクタ単位で行います（注1）。

注1

XCのプログラマーズマニュアルではセクタ単位と記載されています。ですが，FATがクラスタ単位なので，セクタではなくクラスタ（論理セクタ）単位の可能性もあります。さて，今回のSRAMディスクでは1クラスタを1セクタとして扱っているのでこの疑惑を追究できませんでした。後日，HDDのように1クラスタ数セクタの装置を扱うときには要注意の項目です。

また，リクエストヘッダの「開始セクタ」は表中ではロング型（4バイト）となっていますが，アセンブラソース中ではワード（2バイト）として処理しているなど少々気になるところです。今回作ったプログラムではロングで処理していますが，念のため内部で補正をかけ，オリジナルのアセンブラソースと等価にしています。やはり，これもHDDのときにどう変化するかチェックポイントでしょうか。

図5 Human68kとデバイスドライバの制御の流れ

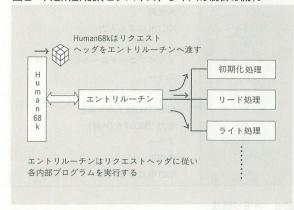



表5 初期化コマンド

コマンドコード＝0

位置	バイト数	I/O	内容
0	1		26（定数）
1	1		未使用
2	1	I	コマンドコード（＝0）
3	1	O	エラーコード（LOW）
4	1	O	エラーコード（HIGH）
5	8		未使用
13	1	O	ユニット数
14	4	O	デバイスドライバの最終アドレス
18	4	I	パラメータポインタ
		O	BPBテーブルのポインタ
22	1	I	ドライブ番号（0＝A，……）

表6 ディスク交換チェックコマンド

コマンドコード＝1

位置	バイト数	I/O	内容
0	1		26（定数）
1	1	I	ユニット番号
2	1	I	コマンドコード（＝1）
3	1	O	エラーコード（LOW）
4	1	O	エラーコード（HIGH）
5	8		未使用
13	1	I	メディアバイト
14	1	O	データバッファ

表7 入出力コマンド

コマンドコード＝4，8，9

位置	バイト数	I/O	内容
0	1		26（定数）
1	1	I	ユニット番号
2	1	I	コマンドコード（＝4，8，9）
3	1	O	エラーコード（LOW）
4	1	O	エラーコード（HIGH）
5	8		未使用
13	1	I	メディアバイト
14	4	I	転送バッファのアドレス
18	4	I	セクタ数
22	4	I	開始セクタ番号

表8 ドライブコントロールコマンド

コマンドコード＝5

位置	バイト数	I/O	内容
0	1		26（定数）
1	1	I	ユニット番号
2	1	I	コマンドコード（＝5）
3	1	O	エラーコード（LOW）
4	1	O	エラーコード（HIGH）
5	8		未使用
13	1	I	内部コマンド
		O	ドライブの状態

内部コマンド

0 ドライブの状態を調べる

1 イジェクトする

2 イジェクト禁止1（ユーザー用）

3 イジェクト許可1（ユーザー用）

4 ディスクがセットされていないとLED点灯

5 ディスクがセットされていないとLED消灯

6 イジェクト禁止2（OS用）

7 イジェクト許可2（OS用）

5) 命令No.5，ドライブコントロール&センス（表8）

ディスクの状態設定および検出を行います。この命令はMS-DOSでは欠番になっています。Human68kでの拡張と思われます。関数dskctrl()で処理します。

実際には，SRAMがRAMディスクとして使用されているかチェックし，使用されていれば，リクエストヘッダの「ドライブの状態」へ42Hのコードを設定します。42_Hは，FDDでいうところのLEDが点灯しメディアが入っている状態を示すコードです。

RAMディスクとして使用されてないときは，02_Hすなわちドライブノットレディのコードをセットします。

アセンブル&コンパイル&リンク方法

最後になってしまいましたが，一応アセンブル&コンパイル&リンクの方法を示しておきます。使用方法は前回のものと同一です。

```
AS D0.S
AS D2.S
CC /Fc D1.C
LK /ODRVR.SYS D0.O D1.O /LIB¥DOSLIB.L D2.O
```

次回予告

ようやく，基礎実験が終了してきました。次回は今まで行ってきた通信，デバイスドライバの実験成果を複合させ，システムにまとめる予定です。

ブロック型デバイスドライバをひとつ作成します。そして，このデバイスドライバには通信機能を内蔵させ，外部のX68000と交信させます。つまりは相手のX68000を仮想ディスクとして働かせます。実験ということで，今回同様SRAMディスクを利用し相手のSRAMディスクを認識させるつもりです。

参考文献

C Compiler PRO-68K ver.2.0プログラマーズマニュアル，シャープ

Cによるメモリ管理技法，中島信行（ただし，本書はMS-DOS用です）

MS-DOSディスク管理技法，中島信行（ただし，本書はMS-DOS用です）

リスト1

```
==================  d0.s  ==================
  1: ****************************************************************
  2: *  DEVICE DRIVER SAMPLE No1
  3: *  XCプログラマーズマニュアル.P643のSRAMDISKプログラム
  4: *  を改造して作りました。
  5: ****************************************************************
  6:     .include        doscall.mac
  7:     .text
  8:
  9: .xref          _dskent
 10: .xref          _fatdir
 11: .xref          _dskini
 12: .xref          _mediac
 13: .xref          _dskctrl
 14: .xref          _dskout
 15: .xref          _dskotv
 16: .xref          _dskinp
 17: .xref          _notcom
 18:
 19: .xdef          _d_dat
 20: .xdef          _d_tim
 21: .xdef          _d_dte
 22: .xdef          _inittbl
 23: .xdef          _rdmaxr
 24:
 25: dsktbl:        dc.l    -1
 26:     dc.w    $0000
 27:     dc.l    dskstr
 28:     dc.l    dskent
 29:     dc.b    1,'S_RAM123'
 30: dskreq:        dc.l    0
 31: dskjmp:        dc.l    _dskini
 32:     dc.l    _mediac
 33:     dc.l    _notcom
 34:     dc.l    _notcom
 35:     dc.l    _dskinp
 36:     dc.l    _dskctrl
 37:     dc.l    _notcom
 38:     dc.l    _notcom
 39:     dc.l    _dskout
 40:     dc.l    _dskotv
 41:     dc.l    _notcom
 42:     dc.l    _notcom
 43:     dc.l    _notcom
 44:
 45: dskstr:        move.l  a5,dskreq
 46:     rts
 47:
 48: dskent:
 49:     movem.l d0-d7/a0-a6,-(sp)          * 各レジスタをバックアップ
 50:
 51:     move.l  dskreq,a5                  * C関数へ引数を渡す
 52:     move.l  a5,-(sp)
 53:     jsr     _dskent                    * C関数をコール
 54:     addq.l  #4,SP                      * 後始末(引数受け渡しで使ったスタックを処分)
 55:
 56:     movem.l (sp)+,d0-d7/a0-a6          * 各レジスタを復旧
 57:     rts                                * Humanへ帰る.さやうならぁ.
 58:
 59: bpbtbl:
 60:     dc.w    1024
 61:     dc.b    1
 62: bpbftc:     dc.b    1
 63:     dc.w    1
 64: bpbdrc:     dc.w    32
 65: _rdmaxr:    dc.w    $10
 66: bpbfid:     dc.b    $f9
 67: bpbfsz:     dc.b    1
 68:
 69: _inittbl:
 70:     dc.l    bpbtbl
 71:
 72: _d_dat:     dc.b    'SRAM_DISK  ',$08,0,0,0,0
 73:     dc.b    0,0,0,0,0,0
 74: _d_tim:     dc.b    $A0,$60
 75: _d_dte:     dc.b    $E7,$0A
 76:     dc.b    0,0,0,0,0,0
 77:
 78:     end
```

リスト2

```
===============  d2.s  ==================
  1: ****************************************************************
  2:     .text
  3:
  4: .xdef      _inittblend
  5:
  6: _inittblend:
  7:     dc.b    0
  8:
  9:     end
```

リスト3

```
===============  d1.c  ==================
  1: #include   <doslib.h>
  2: #include   <stdio.h>
  3:
  4: extern     short   d_tim;
  5: extern     short   d_dte;
  6: extern     char    d_dat[];
  7:
  8: /*
  9: extern     struct  BPBPOI  *inittbl;
 10: */
 11: extern     char    inittbl;
```

```
 12: extern     char    inittblend;
 13: extern     unsigned short  rdmaxr;
 14:
 15: #define    SRAM    0xed0000
 16: #define    SRAMMD  0xed002d
 17: #define    SYS00d  0xe8e00d
 18:
 19: #define    REQLEN  0
 20: #define    UNITCD  1
 21: #define    COMCOD  2
 22: #define    ERRLOW  3
 23: #define    ERRHIGH 4
 24:
 25: #define    MXUNIT  13
 26: #define    DEVEND  14
 27: #define    BPBPOI  18
 28: #define    BDEVNO  22
 29:
 30: #define    DISKID  13
 31: #define    DISKFG  14
 32:
 33: #define    DMAADR  14
 34: #define    DMALEN  18
 35: #define    STAREC  24
 36: #define    GETDAT  13
 37:
 38: char       *_devmes = "C: $ED0400-15K";
 39: char       *_mesini = ")初期化しました ¥n¥r";
 40: char       *_mesnoi = ")初期化しませんでした ¥n¥r";
 41:
 42: void       fatdir();
 43: int        dskinp();
 44: int        dskout();
 45: int        dskini();
 46: int        mediac();
 47: int        dskpaw();
 48: int        dskotv();
 49:
 50: struct REQ_HED {
 51:     char    reqlen;
 52:     char    unitcd;
 53:     char    comcod;
 54:     char    errlow;
 55:     char    errhigh;
 56: };
 57:
 58: struct REQ_INI {
 59:     char    reqlen;             /* リクエストヘッダの長さ        */
 60:     char    unitcd;             /* ユニットコード                */
 61:     char    comcod;             /* コマンドコード                */
 62:     char    errlow;             /* エラーコードその1             */
 63:     char    errhigh;            /* エラーコードその2             */
 64:     char    rsv[8];             /* 予約領域                      */
 65:                                 /* --------------------------    */
 66:     char    mxunit;             /* ユニット数                    */
 67:     char    *devend;            /* デバイスドライバ制御プログラムの終了アドレス */
 68:     char    *bpbpoi;            /*struct        BPBPOI *bpbpoi;  BPBテーブルアドレ
ス          */
 69:     char    bdevno;             /* ブロックデバイス番号(0=A*)    */
 70: };
 71:
 72: struct REQ_CHG {
 73:     char    reqlen;             /* リクエストヘッダの長さ        */
 74:     char    unitcd;             /* ユニットコード                */
 75:     char    comcod;             /* コマンドコード                */
 76:     char    errlow;             /* エラーコードその1             */
 77:     char    errhigh;            /* エラーコードその2             */
 78:     char    rsv[8];             /* 予約領域                      */
 79:                                 /* --------------------------    */
 80:     char    diskid;             /* よくわかんない<現在未使用>    */
 81:     long    diskfg;             /* よくわかんない 資料ではバイト型*/
 82: };
 83:
 84: struct REQ_RW {
 85:     char    reqlen;             /* リクエストヘッダの長さ        */
 86:     char    unitcd;             /* ユニットコード                */
 87:     char    comcod;             /* コマンドコード                */
 88:     char    errlow;             /* エラーコードその1             */
 89:     char    errhigh;            /* エラーコードその2             */
 90:     char    rsv[8];             /* 予約領域                      */
 91:                                 /* --------------------------    */
 92:     char    diskid;             /* よくわかんない<現在未使用>    */
 93:     char    *dmaadr;            /* 転送バッファ                  */
 94:     long    dmalen;             /* 転送セクター数                */
 95:     char    starec00[2];        /* アクセスセクタ番号上位2byte   */
 96:     long    starec;             /* アクセスセクタ番号            */
 97: };
 98:
 99: struct REQ_CTRL {
100:     char    reqlen;             /* リクエストヘッダの長さ        */
101:     char    unitcd;             /* ユニットコード                */
102:     char    comcod;             /* コマンドコード                */
103:     char    errlow;             /* エラーコードその1             */
104:     char    errhigh;            /* エラーコードその2             */
105:     char    rsv[8];             /* 予約領域                      */
106:                                 /* --------------------------    */
107:     char    getdat;
108: };
109:
110: /****************************************************************
111:     diskent 世に言うエントリールーチン (紬来語)
112:     エントリー(entry)は全プログラムの入り口と解釈すること
113:     ASMを強引にハンドディスコンパイル
114: ****************************************************************/
115: void dskent( req_hed )
116: struct REQ_HED *req_hed;
117: {
118:   int     sts;
119:
120:     switch( req_hed->comcod ) {
121:         case 0:
122:                 sts = dskini( req_hed );
123:                 break;
124:         case 1:
125:                 sts = mediac( req_hed );
126:                 break;
127:         case 2:
128:                 sts = notcom();                 /* 未使用命令          */
129:                 break;
130:         case 3:
131:                 sts = notcom();                 /* 未使用命令          */
132:                 break;
133:         case 4:
134:                 sts = dskinp( req_hed );
135:                 break;
136:         case 5:
137:                 sts = dskctrl( req_hed );
138:                 break;
139:         case 6:
140:                 sts = notcom();                 /* 未使用命令          */
141:                 break;
142:         case 7:
143:                 sts = notcom();                 /* 未使用命令          */
144:                 break;
145:         case 8:
146:                 sts = dskout( req_hed );
147:                 break;
148:         case 9:
149:                 sts = dskotv( req_hed );
150:                 break;
151:         case 10:
152:                 sts = notcom();                 /* 未使用命令          */
153:                 break;
154:         case 11:
155:                 sts = notcom();                 /* 未使用命令          */
156:                 break;
157:         case 12:
158:                 sts = notcom();                 /* 未使用命令          */
159:                 break;
160:         default:
161:                 sts = notcom();                 /* 未使用命令          */
162:                 break;
163:     }
164:
165:     req_hed->errlow = (char)(sts & 0x000000ff);     /* 下位バイトエラーコード格納
 */
166:
167:     sts >>= 8;
168:     req_hed->errhigh = (char)(sts & 0x000000ff);    /* 上位バイトエラーコード格納
 */
169:
170: }
171:
172: /****************************************************************
173:     mediac  メディア交換処理(RAMなので未使用のはず)
174:     ASMを強引にハンドディスコンパイル
175: ****************************************************************/
176: int notcom()
177: {
178:     return( 0x5003 );
179: }
180:
181: /****************************************************************
182:     mediac  メディア交換処理(RAMなので未使用のはず)
183:     ASMを強引にハンドディスコンパイル
184: ****************************************************************/
185: int mediac( req_chg )
186: struct REQ_CHG *req_chg;
187: {
188:   char     *adr;
189:
190:     adr = (char*)(SRAM+0x400);
191:
192:     if( *adr=='¥xf9' ) {
193:             req_chg->diskfg = 0L;
194:     }
195:     else {
196:             req_chg->diskfg = 1L;
197:
198:     }
199:
200:     return( 0 );
201: }
202:
203: /****************************************************************
204:     dskinp  ディスク入力処理(?)
205:     ASMを強引にハンドディスコンパイル
206: ****************************************************************/
207: int dskinp( req_rw )
208: struct REQ_RW *req_rw;
209: {
210:   int      sts;
211:   char     *adr;
212:   char     c;
213:
214:     sts = 0x42;
215:
216:     adr = (char*)(SRAM+0x400);
217:     c = '¥xf9';
218:
219:     if( c!=*adr ) {                                 /* RAMDISKのチェック     */
220:             sts = 0x5007;                           /* Err Code セット       */
221:     }
222:     else {
223:             sts = dskpaw( req_rw, 0L );
224:     }
225:
226:     return( sts );
227: }
228:
```


▶昔のゲームを集めたCD-ROM専用のソフトなんか出ればCD-ROMドライブの所有者も増えるかも。
山本　尉(16)千葉県

```
229: /*****************************************************************
230:     dskotv  出力&ベリファイ処理(?)
231:     ASMを強引にハンドディスコンパイル
232: *****************************************************************/
233: int dskotv( req_rw )
234: struct REQ_RW *req_rw;
235: {
236:    int       sts;
237:
238:    sts = dskout( req_rw );
239:    if( sts ) {                        /* ERR ?         */
240:             ;                         /*    Yes.       */
241:    }
242:    else {
243:             sts = dskpaw( req_rw, 2L );
244:    }
245:
246:    return( sts );
247: }
248:
249: /*****************************************************************
250:     dskout  ディスク出力処理(?)
251:     ASMを強引にハンドディスコンパイル
252: *****************************************************************/
253: int dskout( req_rw )
254: struct REQ_RW *req_rw;
255: {
256:    char      *adr;
257:    int       sts;
258:
259:    adr = (char*)(SYS00d);             /* SRAM書込許可          */
260:    *adr = 0x31;                       /*        "             */
261:
262:    sts = dskpaw( req_rw, 1L );
263:
264:    *adr = 0;                          /* SRAM書込禁止セット     */
265:
266:    return( sts );
267: }
268:
269: /*****************************************************************
270:     dskctrl ディスクコントロール
271:     ASMを強引にハンドディスコンパイル
272: *****************************************************************/
273: int dskctrl( req_ctrl )
274: struct REQ_CTRL *req_ctrl;
275: {
276:    char      *adr;
277:
278:    req_ctrl->getdat = 0x42;
279:
280:    adr = (char*)(SRAM+0x400);
281:
282:    if( *adr!='¥xf9' ) {
283:             req_ctrl->getdat = 2;
284:    }
285:
286:    return( 0 );
287: }
288:
289: /*****************************************************************
290:     dskini  ディスク初期化ルーチン
291:     ASMを強引にハンドディスコンパイル
292: *****************************************************************/
293: int dskini( req_ini )
294: struct REQ_INI *req_ini;
295: {
296:    char      *adr;
297:    int       sts;
298:    char      c;
299:
300:    sts = 0x700d;                                  /* Err code set
   */
301:
302:    adr = (char*)(SRAMMD);                         /* SRAM使用モードチェック
 */
303:    if( *adr!=1 ) {                                /* 1=RAM-DISK mode
   */
304:             PRINT( (unsigned char*)"S-RAMをRAM-DISKとして使用できません" );
305:    }
306:    else {
307:             req_ini->mxunit = 1;             /* 最大unit セット           */
308:             req_ini->bpbpoi = &inittbl;            /* BPBテーブルアドレスセット
*/
309:
310:             req_ini->devend = &inittblend;  /* デバイステーブルエンドアドレスセット
 */
311:
312:             *_devmes = req_ini->bdevno  + 'A'; /* デバイス番号取得          */
313:             PRINT( (unsigned char*)_devmes );
314:
315:             adr = (char*)(SRAM+0x400);
316:             c = '¥xf9';
317:
318:             if( *adr!= c ) {                     /* !=0xf9 .初期化する
  */
319:                      fatdir();
320:                      PRINT( (unsigned char*)")初期化しました ¥n¥r" );
321:             }
322:             else if( K_KEYBIT(0x0e)&1 ) {   /* シフトキーが押されていれば無条件初期
化     */
323:                      fatdir();
324:                      PRINT( (unsigned char*)")初期化しました ¥n¥r" );
325:             }
326:             else {
327:                      PRINT( (unsigned char*)")初期化しませんでした ¥n¥r" );
328:             }
329:
330:             sts = 0;                             /* 初期化成功
 */
331:    }
332:
333:    return( sts );
334: }
335:
336: /*****************************************************************
337:     ディレクトリ管理領域初期化
338:     ASMを強引にハンドディスコンパイル
339: *****************************************************************/
340: void fatdir()
341: {
342:    char      *adr;
343:    char      *adr1;
344:    long      *ladr;
345:    int       i;
346:    long      n;
347:
348:    adr = (char*)SYS00d;                /* SRAM書込許可 */
349:    *adr = 0x31;
350:
351:    adr = (char*)(SRAM + 0x400);
352:
353:    for( i=0 ; i<2048 ; i++ ) {         /* fat clr          */
354:             adr[i] = 0;
355:    }
356:
357:    ladr = (long*)adr;
358:    *ladr = 0xf9ffff00;
359:
360:    n = GETTIME();
361:    n >>= 8;
362:    d_tim = n;
363:
364:    n = GETDATE();
365:    n >>= 8;
366:    d_dte = n;
367:
368:    adr = d_dat;
369:    adr1 = (char*)(SRAM + 0x800);
370:    for( i=0 ; i<32 ; i++ ) {
371:             adr1[i] = adr[i];
372:    }
373:
374:    adr = (char*)SYS00d;                /* SRAM書込禁止 */
375:    adr = 0;
376:
377: }
378:
379: /*****************************************************************
380:     dskpaw
381: *****************************************************************/
382: int dskpaw( req_rw, mode )
383: struct REQ_RW        *req_rw;
384: long                 mode;
385: {
386:    int       sts;
387:    char      *a1;
388:    char      *a2;
389:    int       i;
390:    int       no;
391:    long      wk;
392:    wk = req_rw->starec;
393:    wk >>= 16;
394:    sts = wk + (req_rw->dmalen & 0x0000ffff);
395:
396:
397:    if( rdmaxr<sts ) {
398:             sts = 0x5007;
399:    }
400:    else {
401:
402:             sts = req_rw->dmalen;                /* no = 転送バイト数算出*/
403:             sts <<= 10;                          /* no = dmalen * 1024 -1*/
404:             no = sts - 1;
405:
406:             a1 = req_rw->dmaadr;                 /* a1 = 転送バッファアドレス*/
407:
408:             sts = req_rw->starec;                /* a2 = RAMDiskアドレス */
409:             sts &= 0xffff0000;
410:             sts >>= 6;
411:             sts += 1;
412:             a2 = (char*)(SRAM+sts);
413:
414:             if( mode == 0L ) {                   /* DISK読み取り          */
415:                      for( i=0 ; i<no ; i++ ) {
416:                               a1[i] = a2[i];
417:                      }
418:                      sts = 0;
419:             }
420:             else if( mode == 1L ) {              /* DISK 書き取り         */
421:                      for( i=0 ; i<no ; i++ ) {
422:                               a2[i] = a1[i];
423:                      }
424:                      sts = 0;
425:             }
426:             else if( mode == 2L ) {              /* ベリファイ            */
427:                      sts = 0;
428:                      for( i=0 ; i<no ; i++ ) {
429:                               if( a1[i]!=a2[i] ) {
430:                                        sts = 0x700b;
431:                                        break;
432:                               }
433:                      }
434:             }
435:    }
436:
437:    return( sts );
438: }
```

▶「MATIER」のエフェクトは10MHz機だと重い。でもスゴイ。
佐怒賀　英一(26)神奈川県

SIDE A

とりあえず路面をつかまえる

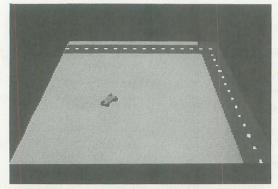
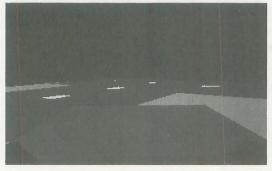
Tan Akihiko 丹 明彦

今回はコーナリングに焦点を絞り，水平面での正しい旋回を示す
より実車に近づけるためにはどのような要素が必要となるのか
疑似3Dカーレースの弱点を暴きながら解説していく

「WORLD CIRCUIT」の事実

さて，いきなりおわびしなくてはならない。付録ディスクは少し延期になった。本連載の成果を皆様のもとにお届けできるのはもう少し先のことになる。しかしいっちゃなんだが，私にとっては好都合であった。数カ月あれば，もう少しましなドライビングシミュレータも作れるというものだ。私のテンションが多少落ちたことは否めないが，それとて煮詰まってしまって変なものを出すよりはましだろう。

*　*　*

AMIGAやIBM互換機で最高のF-1シミュレータとしての評価を得続けている「WORLD CIRCUIT」であるが，先日英国のAMIGA雑誌を読んでいたら作者のインタビューが載っていた。それによると，どうも本物のF-1チームの協力のもとに制作されたらしい。サーキットの異様な精密さなどもこれで説明がつく。

さらには，とあるF3ドライバーのエピソードまで書いてあった。スパーフランコルシャン・サーキットは，ふだんは公道のためレース期間中しか走れないのだが，彼はこのコースを「WORLD CIRCUIT」で練習し，実際のレースでも勝ってしまったというのである。こうした話を真正直に信じこんでしまうほど私もお人好しでないつもりだが，こと「WORLD CIRCUIT」に関する限り，妙な説得力を感じてしまうのだ。それはやはり「WORLD CIRCUIT」がゲームとしての体裁を整えただけのものでなく，シミュレーション性を突き詰めたものだからなのだろう。

今回のプログラム

コックピットからのビュー

腐ったコーナリング

さて，我々は自動車の運動を力学的にシミュレートしようとしているわけだが，その際規範とすべきものを確立しておく必要がある。

まずはっきりさせておきたいが，F-1の車載カメラにそっくりになるように作ろう，というようなアプローチはとらない。

ハンドルを右に切れば車が右に動く，という上っ面のみを見ながら多くの疑似3Dカーレースゲームは制作されてきた。典型的なラスターもののレーシングゲームのデザインの一例を挙げよう(図1)。高速コーナーでは道路が緩やかに長い間曲がっている。逆にシケインやヘアピンは道路がきつくはあるが瞬間的にしか曲がっていない。なるほど確かに車載カメラの映像はそうなっている。しかしこれを「ハンドルの切れ角＝X方向の単位時間あたりの移動量」などというデザインにしたとたん悲劇は起こる。ラスターもののレーシングゲームでは，超高速コーナーと呼ばれているコーナーよりもシケインのほうが

高速にクリアできるという奇怪な現象が現実に起きているのだ。そしてこれを抑えるために，無根拠でわけのわからないパラメータを導入し，バランスをとるべく調整に追われることになるのだろう。これではいけないのだ。

きちんと車の挙動の本質を理解し，計算速度の制限と戦いながら，これだけは譲れないというところをきちんとモデル化する。そうすれば結果は自然についてくる。結果的に車載カメラの映像そっくりになってもいっこうにかまわない，むしろそうなってもらわなくては困る。

この方針決定は自分自身に対する宣言でもある。

車は走り，曲がり，止まる機械だ

そこで1冊の本を紹介しておこう。しばらく前のOh!X(参考文献1)で華門真人氏が紹介していた『クルマの速さが見えてきた』という本(参考文献2)である。ドライビングテクニック教授本みたいなタイトルだが，中身は大違い。クルマの速さというものを極めて明晰な理論でもって分析している。これが実践できれば車を思いのままにドライブできるに違いない。といっても自動車力学の観点から難解な微分方程式のオンパレードというものでもない。文章は平易で読みやすい。そして著者の舘内端氏が車というものの本質をつかまえていることを実感させてくれるのだ。

“クルマの動きには，走り，曲がり，止まるしかない”

この言葉は，その『クルマの速さが見えてきた』からの受け売りである。アクセルを踏めば走り，ハンドルを切れば曲がり，ブレーキを踏めば止まる。車を運転するというのはこの作業の繰り返しなのである。

当たり前と思うなかれ。この当然とも思える基本を守っていないカーレースゲームのどれほど多いことか。いみじくも中野修一氏が語った(参考文献3)ように「車を左右に動かして障害物を避けながらできるだけ車をコース内に入れてください」という，車の自然な挙動とはおよそ縁遠いゲームが世の中に溢れかえっているではないか。きちんと空間を，路面をつかまえてドライブせよとこの言葉は教えているのだ。

とまあ，『クルマの速さが見えてきた』は，このほかにも数々の味のある示唆に溢れている。自動車力学を極め，レースの経験も積み，クルマの本質に迫った者のみが知りえる秘密を明かしてくれる貴重な本なのである。

もちろん，堅い数式を使った力学シミュレーションもしなくてはならない。自動車力学の教科書として私は参考文献4を用いていくつもりだ。そんなときでも，『クルマの速さが見えてきた』は，物理現象の数値的解析という枝葉末節にとらわれそうになったときの心のよりどころとして大切な存在である。ドライブシミュレータの制作を志す者はこの書をバイブルとして携えておかなくてはなるまい。

とりあえず旋回運動

さて，今回は先月までの円形サーキットをひとまず忘れ，チープな舞台でシケた車を操縦することにする（図2）。

まず舞台であるが，とりあえず水平面とする。つまり傾きはない。これにより，計算は若干楽になっている。いずれはアップダウンやバンクにも手をつける。

次に車であるが，箱5つのなんともいいかげんな簡易自動車を作る。これはひとえにプログラム行数

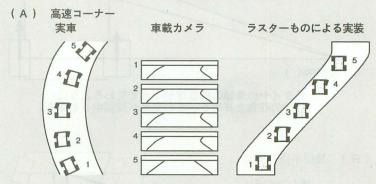
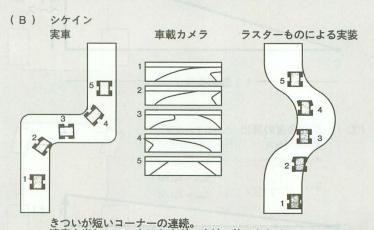
図1　ラスターものの弱点

を抑えるための処置である。先月号で紹介したフォーミュラマシンはさらに改良を加えたが，そのおかげでプログラムは1000行を超えてしまった。さすがにこれを載せるわけにはいかない。その代わりといってはなんだが，ステアリングの基本を押さえる意味でも，ハンドルを切れば前輪が曲がるようになっている。

図2　今回の舞台（水平なフィールド）

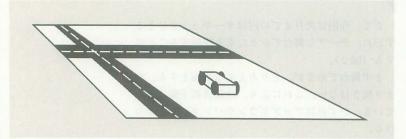
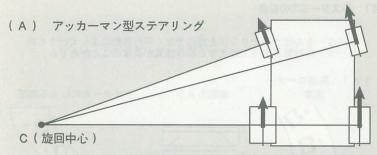
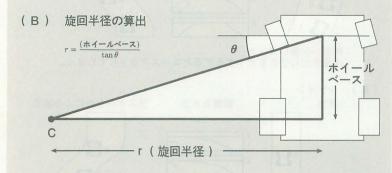
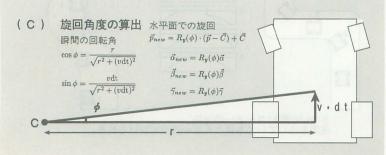



図3　極低速定常円旋回

今回扱う車の運動は，「直進運動」および「極低速定常円旋回運動」と呼ばれる運動である。後者は，ハンドルを切ってごく遅い速度で曲がる運動である。遠心力も無視できる程度のゆっくりとした旋回運動である。そのほか，路面とタイヤの摩擦や空気抵抗など，あらゆる自然界の法則を無視している。純粋に走り，曲がり，止まるという動作だけを行う。そのかわり，タイヤが路面をグリップするという部分だけは押さえてある。

直進運動については詳しい説明の必要はないだろう。速度に前回描画時からの経過時間を掛けると移動量を求めることができる。これを車体の位置ベクトルに足せばよい。

先月号でちょっと触れた経過時間の測定を使って，時間管理を行っている。これを用いれば，速い計算機で実行しても遅い計算機で実行しても車の速度は変わらない。計算機が速ければ速いほど車の動きが滑らかになっていく。これは速度差のある計算機が混在する時代のプログラムとしては必須の処理である。ただし，将来的にはカーレースゲームの通信対戦なども考えており，そうなると時間管理は少し面倒になる。表示はともかく，操縦や力学的動作は両者で同期をとらなくてはならない。

そして今回のメインの旋回運動である。旋回運動とは，ハンドルを切って車を進行させた場合に，ある点を中心として回転する運動である。もちろん，このとき，車体の向きも変わる。

世の中の自動車の操舵システムは，図3(A)の「アッカーマン型ステアリング」になっている。自動車は4本のタイヤで接地している。駆動輪（いわゆるFR車の場合は後輪，FF車の場合は前輪）から力が地面に伝わり，地面からの反力により車体が前進する。このときの進行方向は，舵輪（たいていは前輪）の角度によって決まる。ハンドルを切っておくと，4本のタイヤの向きによって最も自然な方向に車体は回頭する。さて，アッカーマン型ステアリングは，舵輪の角度が左右で微妙に異なる。どのようにハンドルを切っても，4本のタイヤの車軸の延長線が常に1点（これが旋回中心となる）で交わるような操舵機構である。こうすることで，4本のタイヤの進行方向が旋回円の接線方向と常に一致することが保証され，無理なく路面をとらえることができるのだ。

もちろん，このような機構は現実の力学的な要求から生じたもので，コンピュータで忠実にシミュレートする必要はない。今回のプログラムでも前輪の舵角を左右で同じにして表示している。が，あくまでリアリティを追求するなら，前輪の表示を左右で微妙に違わせることで，もしかしたらよりグリップ感が増すかもしれない。

今回は，もう少しモデルを単純化し，車体の中央

に仮想的な前輪と後輪を設け，その2輪によって決定される運動を計算している。まず前輪の切れ角とホイールベース（前後輪の間隔）から旋回半径（後輪から旋回中心までの距離）を求め(図3(B))，微小時間（今回のプログラムでは1回の処理ループにかかる時間ということになるので数分の1秒）で車の進む距離と旋回半径からその微小時間に車の旋回する角度を求めている(図3(C))。

今月のプログラム

水平面フィールドの上で自動車が走り，曲がり，止まるだけのプログラムである。

マウスで操作する。マウスを左右に動かすとハンドルの切れ角が変化する。マウスの右ボタンを押せば加速し，左ボタンで減速する。F1キーを押すことで，視点を車外と車上で切り替えられる。プログラム起動時の視点は車外にある。操縦に慣れたら視点を切り替えてみるといいだろう。

ロードホールディング以外はまったくいいかげんである。等加速度運動（エンジン特性などおかまいなし）だし，最高速度は無制限，遠心力もない。さらにはフィールドの外にはみ出しても，車が転落することもない。

もちろん，それらしいパラメータを導入すればいいのはわかっている。たとえば最高速というパラメータを設けて，加速処理の部分に比較のためのコードを1行入れれば最高速度が制限できる。でもそういうパラメータ調整をいまやってもしかたがない。いずれ空気抵抗などの要素を力学シミュレーションに組み込めば，最高速は自然と頭打ちになるはずである。ゆえにいいかげんとわかっている部分も故意に放置してある。

*　　*　　*

付録ディスクが延期されたこともあるので，今回は完結したプログラムを掲載したい。

まず1993年10月号の付録ディスクに収録したSLASH開発キットを用意する。コンパイルは終了しておくことが必要だ。そのトップディレクトリに「9401」という名前のディレクトリを作り，前回掲載のeulerlib.hおよびeulerlib.cを入れる。次に今回掲載するtimedifference.h，dtest.cおよびMakefileを作成して同じディレクトリに入れる。まとめると次のようになる。

LIB¥	SLASH開発キット
9401¥eulerlib.c	前回掲載
9401¥eulerlib.h	前回掲載
9401¥timedifference.h	今回掲載
9401¥dtest.c	今回掲載
9401¥Makefile	今回掲載

そして9401ディレクトリに移ってMAKEを実行し，コンパイルが成功すればdtest.xという実行ファイルができあがる。

今後の予定

ここしばらくはこのプログラムを拡張していく予定である。車の緒元を収めたCARINFO構造体は今後もメンバを増やしていくことだろう。

まず水平面での運動を完璧にしてから斜めの路面に応用するか，または早めに斜めの路面を導入してその上で自動車の運動を極めるかはこれから決めようと考えている。それではまた来月。

■参考文献

1：シミュレーションプログラミング入門（第3回）シミュレーションは未来を開く，華門真人，Oh!X1991年2月号pp.58-63，ソフトバンク(1991)
2：クルマの速さが見えてきた 「ドライビング・ハイ」を生むメカニズム，舘内端，光文社(1989)
3：SLASHに寄せて，中野修一，Oh!X1993年10月号pp.57-58，ソフトバンク(1993)
4：自動車力学，景山克三/景山一郎，理工図書(1984)

■リスト1　MAKEFILE

```
SLASHLIBDIR = ..¥lib
COLORDIR = ..¥color
CCOPTS = -O -Wall

%.o: %.c
	gcc $(CCOPTS) -c $<

all: dtest.x

dtest.x: dtest.o eulerlib.o
	gcc -o dtest.x dtest.o eulerlib.o¥
		$(COLORDIR)¥tpllib.a¥
		$(SLASHLIBDIR)¥_slashlib.a¥
		$(SLASHLIBDIR)¥_utillib.a¥
		$(SLASHLIBDIR)¥slashlib.a

dtest.o: dtest.c eulerlib.h timedifference.h
eulerlib.o: eulerlib.c eulerlib.h

clean:
	if exist *.dat del -y *.dat
	if exist *.bak del -y *.bak
	if exist dtest.o del dtest.o
	if exist eulerlib.o del eulerlib.o

distclean: clean
	if exist dtest.x del dtest.x
```

■リスト2　TIMEDIFFERENCE.H

```
 1: /*
 2:         timedifference.h
 3:         - 時間管理のための 1/100 単位の時間計測
 4:           Oct. 1993      丹 明彦(Oh!X)
 5: */
 6:
 7: #ifndef __TIMEDIFFERENCE_H__
 8: #define __TIMEDIFFERENCE_H__
 9:
10: /* IOCSLIB の ONTIME() で計った2時刻の差( = T1 - T2 )
11:    ONTIME() の値が1日を超えるとリセットされることも考慮ずみ
12:    IOCS コールの ONTIME は日数も返すがとりあえず無視している */
13: #define TIMEDIFFERENCE(T1,T2)    (((T1)+8640000-(T2))%8640000)
14:
15: #endif   /* __TIMEDIFFERENCE_H__ */
```

■リスト3　DTEST.C

```
  1: /*
  2:   dtest.c
  3:   - 車の動作
  4:     Nov. 1993  丹 明彦(Oh!X)
  5: */
  6:
  7: #define    __IOCS_INLINE__
  8: #include  <iocslib.h>
  9: #define    __DOS_INLINE__
 10: #include  <doslib.h>
 11: #include  <stdio.h>
 12: #include  <stdlib.h>
 13:
 14: #include  "..¥lib¥_slashlib.h"
 15: #include  "..¥color¥_tpllib.h"
 16: #include  "eulerlib.h"
 17: #include  "timedifference.h"
 18:
 19: /* シェーディングなしにする場合などに用いる色を求める簡易マクロ */
 20: #define  RGBI(R,G,B,I)    (((G)<<11)|((R)<<6)|((B)<<1)|(I))
 21: #define  RGBILONG(R,G,B,I)  ((((G)<<27)|((R)<<22)|((B)<<17)|((
I)<<16))|(((G)<<11)|((R)<<6)|((B)<<1)|(I)))
 22:
 23: #define  N_POINT    800
 24: #define  N_OBJECT  4
 25:
 26: SLPOLYGONLIST  *field_polygonlist;
 27: SLPOINTLIST  *field_pointlist;
 28:
 29: SLPOLYGONLIST  *car_polygonlist;
 30: SLPOINTLIST  *car_pointlist;
 31: SLPOLYGONLIST  *fwheel_polygonlist;
 32: SLPOINTLIST  *fwheel_pointlist;
 33:
 34: SLTRANSWORK  *work;
 35: SLMINMAX  *minmaxfull, *minmaxt;
 36: SLPARAMETER  parameter;
 37:
 38: typedef struct {
 39:   int  width;     /* ボディ全幅 */
 40:   int  length;     /* ボディ全長 */
 41:   int  height;     /* ボディ全高 */
 42:   int  fshaft;     /* Fホイール幅 */
 43:   int  rshaft;     /* Rホイール幅 */
 44:   int  fwidth;     /* Fタイヤ幅 */
 45:   int  fradius;  /* Fタイヤ半径 */
 46:   int  rwidth;     /* Rタイヤ幅 */
 47:   int  rradius;  /* Rタイヤ半径 */
 48:   int  wheelbase;  /* ホイールベース */
 49:   int  steeringratio;  /* ステアリング係数(マウスX座標と回転角の関係) */
 50:   int  accel;     /* 加速度 */
 51:   int  brake;     /* 減速度 */
 52: } CarInfo;
 53:
 54: /* 車 */
 55: void  create_car( c )
 56: CarInfo  *c;
 57: {
 58:   car_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SLPOL
YGON)*18 );
 59:   car_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOINT)*
24 );
 60:   car_polygonlist->n = 0;
 61:   car_pointlist->n = 0;
 62:   /* 車体 */
 63:   makebox( car_polygonlist, car_pointlist,
 64:       -c->width/2, -c->height, -c->length/2,
 65:        c->width/2, 0,          c->length/2,
 66:        &std_red );
 67:   /* 後輪(車体に固定されているため同一のポリゴンリストに入れる) */
 68:   makebox( car_polygonlist, car_pointlist,
 69:       -c->fshaft/2-c->rwidth/2, -c->height, -c->wheelbase/2-c-
>rradius,
 70:       -c->fshaft/2+c->rwidth/2, 0,          -c->wheelbase/2+c-
>rradius,
 71:       &std_darkgray );
 72:   makebox( car_polygonlist, car_pointlist,
 73:       c->fshaft/2-c->rwidth/2, -c->rradius*2, -c->wheelbase/2-
c->rradius,
 74:       c->fshaft/2+c->rwidth/2, 0,            -c->wheelbase/2+
c->rradius,
 75:       &std_darkgray );
 76:   AddNorm( car_polygonlist, car_pointlist );
 77:   SortPoly( car_polygonlist, car_pointlist );
 78:   /* 前輪(車体から独立しているため別のポリゴンリストに入れる) */
 79:   fwheel_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SL
POLYGON)*6 );
 80:   fwheel_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOIN
T)*8 );
 81:   fwheel_polygonlist->n = 0;
 82:   fwheel_pointlist->n = 0;
 83:   makebox( fwheel_polygonlist, fwheel_pointlist,
 84:       -c->rwidth/2, -c->fradius*2, -c->fradius,
 85:        c->rwidth/2, 0,             c->fradius,
 86:        &std_darkgray );
 87:   AddNorm( fwheel_polygonlist, fwheel_pointlist );
 88:   return;
 89: }
 90:
 91: void  destroy_car()
 92: {
 93:   free( car_polygonlist );
 94:   free( car_pointlist );
 95:   free( fwheel_polygonlist );
 96:   free( fwheel_pointlist );
 97:   return;
 98: }
 99:
100:
101: /* フィールド */
102: void  create_field()
103: {
104:   int  x, z, i, j;
105:
106: #define  FS  2048     /* フィールドのサイズ */
107: #define  FD  8     /* フィールドの分割数 */
108: #define  RW  (FS/8)     /* 道路の幅 */
109: #define  LW  (FS/128)  /* センターラインの幅 */
110: #define  LN  16     /* センターラインの数 */
111: #define  LL  (FS/(LN*2))  /* センターラインの長さ */
112:
113:   field_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SLP
OLYGON)*200 );
114:   field_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOINT
)*800 );
115:   field_polygonlist->n = 0;
116:   field_pointlist->n = 0;
117:
118:   for ( i = 0; i < FD; i++ ) {
119:     x = -FS + i*(FS*2/FD);
120:     for ( j = 0; j < FD; j++ ) {
121:       z = -FS + j*(FS*2/FD);
122:       addtetragon( field_polygonlist, field_pointlist,
123:         x,            0, z,
124:         x,            0, z+(FS*2/FD),
125:         x+(FS*2/FD), 0, z+(FS*2/FD),
126:         x+(FS*2/FD), 0, z,
127:         &std_yellow );
128:     }
129:   }
130:
131:   x = rand()%(FS*2)-FS;
132:   addtetragon( field_polygonlist, field_pointlist,
133:     x-RW, 0, -FS, x-RW, 0, FS, x+RW, 0, FS, x+RW, 0, -FS, &std
_darkblue );
134:   for ( i = 0; i < LN; i++ ) {
135:     z = -FS + i*(FS*2)/LN;
136:     addtetragon( field_polygonlist, field_pointlist,
137:       x-LW, 0, z, x-LW, 0, z+LL, x+LW, 0, z+LL, x+LW, 0, z, &s
td_white );
138:   }
139:
140:   z = rand()%(FS*2)-FS;
141:   addtetragon( field_polygonlist, field_pointlist,
142:     -FS, 0, z-RW, -FS, 0, z+RW, FS, 0, z+RW, FS, 0, z-RW, &std
_darkblue );
143:   for ( i = 0; i < LN; i++ ) {
144:     x = -FS + i*(FS*2)/LN;
145:     addtetragon( field_polygonlist, field_pointlist,
146:       x, 0, z-LW, x, 0, z+LW, x+LL, 0, z+LW, x+LL, 0, z-LW, &s
td_white );
147:   }
148:
149:   AddNorm( field_polygonlist, field_pointlist );
150:
151:   return;
152: }
153:
154: void  destroy_field()
155: {
156:   free( field_polygonlist );
157:   free( field_pointlist );
158:   return;
159: }
160:
161: int  main()
162: {
163:   double  radius, centerx, centerz, run, co, si;
164:   int  sp, time = 0;
165:   int  mscur, x, y, msdt, lb, rb;
166:   double  theta = 0.0;  /* ステアリング角 */
167:   VECTOR3  v, va, vb, vc;  /* 視点の位置と基底ベクトル */
168:   VECTOR3  b, ba, bb, bc;  /* 車体の位置と基底ベクトル */
169:   VECTOR3  w, wa, wb, wc;  /* 前輪の位置と基底ベクトル */
170:   VECTOR3  l;     /* 光源の方向ベクトル */
171:   int  t1, t2, dt;  /* 時刻と時差 */
172:   int  velocity = 0;  /* 速度 */
173:   int  viewmode = 0;  /* 視点モード(デフォルトは車外) */
174:
175:   /* 車の情報 */
176:   static CarInfo  testcar = {
177:     100, 300, 60, 150, 160, 20, 20, 30, 30,
178:     220, 4, 1, 2 };
179:
180:   create_field();
181:   create_car( &testcar );
182:
183:   work = malloc( sizeof(SLTRANSWORK)*N_POINT );
184:   minmaxfull = malloc( sizeof(SLMINMAX)*2 );
185:   minmaxt = malloc( sizeof(SLMINMAX)*2 );
186:
187:   /* いつでも全画面クリアする */
188:   minmaxfull[0].xmin = 0;
189:   minmaxfull[0].ymin = 0;
190:   minmaxfull[0].xmax = 255;
191:   minmaxfull[0].ymax = 255;
```

▶1GバイトのHDDを買った。たしかに200Mバイトは狭いと思う。
本田　宗生(24)愛知県

```
192:    minmaxfull[1].xmin = -1;
193:    minmaxfull[1].ymin = -1;
194:    minmaxfull[1].xmax = -1;
195:    minmaxfull[1].ymax = -1;
196:
197:    CRTMOD( 14 );
198:    G_CLR_ON();
199:    B_CUROFF();
200:    MS_INIT();
201:    MS_CUROF();
202:    MS_LIMIT(0,0,255,255);
203:    MS_CURST(128,128);
204:    MS_CUROF();
205:    SKEY_MOD(0,0,0);
206:    sp = SUPER( 0 );
207:
208:    SetClearColor( RGBILONG(0,0,8,0) );
209:    SetWindowSize( 256, 256 );
210:    SetWindowCenter( 128, 128 );
211:
212:    /* 車体 */
213:    b[0]  = 0.0; b[1]  = 0.0; b[2]  = 0.0;   /* 位置(ワールド座標) */
214:    ba[0] = 1.0; ba[1] = 0.0; ba[2] = 0.0;   /* 基底ベクトル(α軸) */
215:    bb[0] = 0.0; bb[1] = 1.0; bb[2] = 0.0;   /* 基底ベクトル(β軸) */
216:    bc[0] = 0.0; bc[1] = 0.0; bc[2] = 1.0;   /* 基底ベクトル(γ軸) */
217:
218:    /* 経過時間を計測する */
219:    t1 = ONTIME();
220:
221:    /* メインループ */
222:    for (;;) {
223:      /* マウス入力 */
224:      mscur = MS_CURGT();
225:      x = mscur/65536;
226:      y = mscur%65536;
227:      msdt = MS_GETDT()%65536;
228:      lb = msdt/256;
229:      rb = msdt%256;
230:      /* ESCキーで終了 */
231:      if ( BITSNS(0x00)&2 ) break;
232:      /* F1で視点モード切り替え */
233:      if ( BITSNS(0x0C)&8 ) {
234:        while ( BITSNS(0x0C)&8 );
235:        viewmode = 1 - viewmode;
236:      }
237:
238:      /* 前回の時刻との差 */
239:      t2 = ONTIME();
240:      dt = TIMEDIFFERENCE(t2,t1);
241:      t1 = t2;
242:      /* 速度 */
243:      if ( rb ) velocity += testcar.accel;
244:      if ( lb ) velocity -= testcar.brake;
245:      if ( velocity < 0 ) velocity = 0;
246:      /* 速度に前回の時刻との差をかければ移動量が出る */
247:      run = velocity*dt;
248:
249:      if ( time%2 == 0 ) {
250:        SetWritePlane( (unsigned short *)0xC00200 );
251:        SetClearPlane( (unsigned short *)0xC00200 );
252:      } else {
253:        SetWritePlane( (unsigned short *)0xC00000 );
254:        SetClearPlane( (unsigned short *)0xC00000 );
255:      }
256:        ClearBox( minmaxfull );
257:
258:      /* 水平面(xz平面)上での極低速円旋回 */
259:      theta = ITOD((128-x)*testcar.steeringratio);  /* ステアリング
角 */
260:      if ( x < 128 ) {  /* 左折 */
261:        double  tmpx, tmpz;
262:        radius = testcar.wheelbase / tan( theta );  /* 旋回半径 */
263:        centerx = b[0] - bc[0]*testcar.wheelbase/2  /* 旋回中心 */
264:                  - ba[0]*radius;
265:        centerz = b[2] - bc[2]*testcar.wheelbase/2
266:                  - ba[2]*radius;
267:        /* 旋回中心を中心として車体を回転する */
268:        co = radius / sqrt( radius*radius + run*run );
269:        si = -run / sqrt( radius*radius + run*run );
270:        /* 位置ベクトルは旋回中心からの相対位置ベクトルを回転する */
271:        b[0] =  (b[0] - centerx)*co + (b[2] - centerz)*si + cent
erx;
272:        b[2] = -(b[0] - centerx)*si + (b[2] - centerz)*co + cent
erz;
273:        /* 基底ベクトルはそのものを回転する */
274:        tmpx = ba[0]; tmpz = ba[2];
275:        ba[0] =   tmpx * co + tmpz * si;
276:        ba[2] = - tmpx * si + tmpz * co;
277:        tmpx = bc[0]; tmpz = bc[2];
278:        bc[0] =   tmpx * co + tmpz * si;
279:        bc[2] = - tmpx * si + tmpz * co;
280:      } else if ( x > 128 ) {  /* 右折 */
281:        double  tmpx, tmpz;
282:        radius = testcar.wheelbase / tan( -theta );
283:        centerx = b[0] - bc[0]*testcar.wheelbase/2
284:                  + ba[0] * radius;
285:        centerz = b[2] - bc[2]*testcar.wheelbase/2
286:                  + ba[2] * radius;
287:        co = radius / sqrt( radius*radius + run*run );
288:        si = run / sqrt( radius*radius + run*run );
289:        b[0] =  (b[0] - centerx)*co + (b[2] - centerz)*si + cent
erx;
290:        b[2] = -(b[0] - centerx)*si + (b[2] - centerz)*co + cent
erz;
291:        tmpx = ba[0]; tmpz = ba[2];
292:        ba[0] =   tmpx * co + tmpz * si;
293:        ba[2] = - tmpx * si + tmpz * co;
294:        tmpx = bc[0]; tmpz = bc[2];
295:        bc[0] =   tmpx * co + tmpz * si;
296:        bc[2] = - tmpx * si + tmpz * co;
297:      } else {  /* 直進 */
298:        b[0] += bc[0] * run;
299:        b[2] += bc[2] * run;
300:      }
301:
302:      /* 視点/視線 */
303:      if ( viewmode == 0 ) {
304:        v[0] =  0.0; v[1]  = -3000.0;        v[2]  = -4000.0;
 /* 位置 */
305:        va[0] = 1.0; va[1] = 0.0;            va[2] = 0.0;      /*
α軸 */
306:        vb[0] = 0.0; vb[1] = sqrt(2.0)/2.0; vb[2] = -sqrt(2.0)/2
.0; /* β軸 */
307:        vc[0] = 0.0; vc[1] = sqrt(2.0)/2.0; vc[2] = sqrt(2.0)/2.
0;  /* γ軸 */
308:      } else {
309:        /* 位置 */
310:        v[0] =  b[0] - (testcar.height*1.5)*bb[0];
311:        v[1]  = b[1] - (testcar.height*1.5)*bb[1];
312:        v[2]  = b[2] - (testcar.height*1.5)*bb[2];
313:        va[0] = ba[0]; va[1] = ba[1]; va[2] = ba[2];  /* α軸 */
314:        vb[0] = bb[0]; vb[1] = bb[1]; vb[2] = bb[2];  /* β軸 */
315:        vc[0] = bc[0]; vc[1] = bc[1]; vc[2] = bc[2];  /* γ軸 */
316:      }
317:      /* 光源の方向ベクトル */
318:      l[0] = -1.0; l[1] = -3.0; l[2] = -2.0;
319:      /* 地面の表示 */
320:      eulerA2G( &parameter, &v, &va, &vb, &vc, &l );
321:      TranslateAll( &parameter, work, field_pointlist, minmaxt )
;
322:      DisplayPolygonList( field_polygonlist, work, minmaxt );
323:
324:      /* 車体の表示 */
325:      eulerA2A( &parameter, &v, &va, &vb, &vc, &b, &ba, &bb, &bc
 );
326:      TranslateAll( &parameter, work, car_pointlist, minmaxt );
327:      DisplayPolygonList( car_polygonlist, work, minmaxt );
328:
329:      /* 前輪 */
330:      /* α軸 */
331:      wa[0] = ba[0]*cos(theta) + bc[0]*sin(theta);
332:      wa[1] = ba[1]*cos(theta) + bc[1]*sin(theta);
333:      wa[2] = ba[2]*cos(theta) + bc[2]*sin(theta);
334:      /* β軸 */
335:      wb[0] = bb[0];
336:      wb[1] = bb[1];
337:      wb[2] = bb[2];
338:      /* γ軸 */
339:      wc[0] = -ba[0]*sin(theta) + bc[0]*cos(theta);
340:      wc[1] = -ba[1]*sin(theta) + bc[1]*cos(theta);
341:      wc[2] = -ba[2]*sin(theta) + bc[2]*cos(theta);
342:      /* 位置 */
343:      w[0] =  b[0] + ba[0]*testcar.fshaft/2 + bc[0]*testcar.whee
lbase/2;
344:      w[1] =  b[1] + ba[1]*testcar.fshaft/2 + bc[1]*testcar.whee
lbase/2;
345:      w[2] =  b[2] + ba[2]*testcar.fshaft/2 + bc[2]*testcar.whee
lbase/2;
346:      /* 右前輪の表示 */
347:      eulerA2A( &parameter, &v, &va, &vb, &vc, &w, &wa, &wb, &wc
 );
348:      TranslateAll( &parameter, work, fwheel_pointlist, minmaxt
);
349:      DisplayPolygonList( fwheel_polygonlist, work, minmaxt );
350:
351:      /* 左前輪 */
352:      /* 位置 */
353:      w[0] =  b[0] - ba[0]*testcar.fshaft/2 + bc[0]*testcar.whee
lbase/2;
354:      w[1] =  b[1] - ba[1]*testcar.fshaft/2 + bc[1]*testcar.whee
lbase/2;
355:      w[2] =  b[2] - ba[2]*testcar.fshaft/2 + bc[2]*testcar.whee
lbase/2;
356:      /* 左前輪の表示 */
357:      eulerA2A( &parameter, &v, &va, &vb, &vc, &w, &wa, &wb, &wc
 );
358:      TranslateAll( &parameter, work, fwheel_pointlist, minmaxt
);
359:      DisplayPolygonList( fwheel_polygonlist, work, minmaxt );
360:
361:      if ( time%2 == 0 ) HOME( 0, 256, 0 );
362:        else     HOME( 0, 0, 0 );
363:      time++;
364:    }
365:    SUPER( sp );
366:
367:    free( work );
368:    free( minmaxt );
369:    free( minmaxfull );
370:    destroy_car();
371:    destroy_field();
372:
373:    B_CURON();
374:    CRTMOD( 16 );
375:    KFLUSHIO( 0xFF );
376:
377:    return 0;
378: }
```

▶おなかがすくと怒りっぽくなるんだよな……。　　服部　直幸(20)広島県

SIDE B

座標系の束縛を叩き込め

Yokouchi Takeshi 横内 威至

SLASHで空間内を自由に飛び回るための問題点はどこにあるのか
その鍵を握るのが座標系である
この座標系を見直し，より正しい3Dシステムを目指す

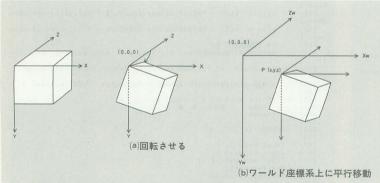
いままで，急ぎ足で説明を続けてきたため，いろいろと不備があったかもしれない。内容も比較的地味であったため，いまいちキャッチーなものではなかったが，3Dの基礎ということで引き続き解説をしていくことにする。なんといってもアセンブルリストを掲載するとなると膨大な量になってしまうため，いまいち説明不足となってしまう。まあ，いままで説明したことはかなりSLASHの内容に近いから，付録ディスクに収録したリストが最大のサンプルと思っていただければ幸いである。

今回からは座標系について考えていきたい。かなり複雑なものであり，いったいどのような座標系及び変換を考えればよいかはいまいちはっきりしない。まず基本的な方法から考えてみよう。

座標系について

すでに丹氏が解説しているように，3次元空間を制御するためには2つの座標系を考えなければならない。物体座標系とワールド座標系の2つだ。描画の際，単純に物体，空間，そしてカメラを考えよう。画面に現れるまでには，本来2段階の手順を考えなければならない。基準とする座標系は空間，つまりワールド座標系である。物体はワールド座標系を基準として姿勢を決定し，そしてワールド座標系内に配置される。次にカメラがワールド座標系を見るのである。

物体の姿勢を決定するのも，ワールド座標系をカメラが見るのも，単純に考えればX，Y，Z軸の3軸に対する回転である。3軸といっても，どのように構成され，回転によってどのような影響が出るのであろうか。順番に考えていこう。

図1

物体の回転は以上のように行う。各点P'n＝P＋R・Pnとなる。
Rは変換行列，Pは物体の位置，Pnは点nの物体座標系を示す。

このとき，(a)と(b)の動作を入れ替える，つまりP'＝R(P＋Pn)とすれば
回転中心－P＝(−x，−y，−z)を中心とした回転ということになる。
地上に固定されている物体などはこのような変換をすることになる。

物体の回転，配置

物体をワールド座標系に配置するためには，
・物体を物体座標系で回転させる
・そのままワールド座標系上に平行移動させる
という動作によって行う(図1)。

すでにわかっていると思うが，物体を回転させるためには変換行列を使って座標変換を行う。この行列は図2のとおり。あまりに有名なため，こう何度も書くのは恥ずかしいが勘弁してもらいたい。本来これに平行移動も組み込み，動作をひとまとめにするため4×4行列にしてもかまわないが，アセンブラサイドから説明を行うために3×3行列にする。なぜなら，平行移動は足し算で行ったほうが断然高速であるからだ。

で，話を戻すが，この3つの行列で変換を行えばX，Y，Z座標，3軸に対して回転を行える。ただし，これらの行列をひとまとめにしておいても座標軸は回転しないことに注意。

さて，ここで物体の回転を考えるとき，決して忘れてはならないオイラー角について説明をしておかねばならない。これは図3に示す。ただし一般的で

はなくSLASHのオイラー角と思ってもらいたい。博識な読者ならばこの説明はオイラー角ではないことに気づくであろう。一般的な航空力学を参考にしてあるが，まず軸の定義が違っている。これは単に入れ替えてあるだけである。しかしそれを除いてもやはり違っていると思うであろう。だがこれで正しいのだ。とりあえず，理由については後述する。

では，ここまでをアセンブラサイドで考える。このあたりのちょっとしたことは図4に示す。まず基本として，3軸の回転行列はあらかじめ掛けておき，1つの行列にしてしまう。座標自体はこの行列で1回変換を行うだけである。行列では三角関数を使っているので，当然小数計算が必要となってくる。となれば基本の16ビット固定小数点を使うことは明白。三角関数はサインテーブルを用意すればいい。三角関数は\$4000〜ー\$4000の範囲で用意する。座標計算後，4倍すればちょうどよくなるためである。

そしてここでのテクニックであるが，11月号に紹

図2

X軸回り

$$Rx=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\theta x & -\sin\theta x\\ 0 & \sin\theta x & \cos\theta x\end{pmatrix}$$

Y軸回り

$$Ry=\begin{pmatrix}\cos\theta y & 0 & \sin\theta y\\ 0 & 1 & 0\\ -\sin\theta y & 0 & \cos\theta y\end{pmatrix}$$

Z軸回り

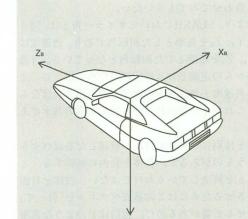
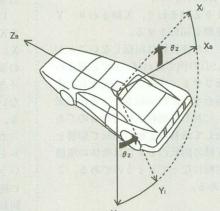
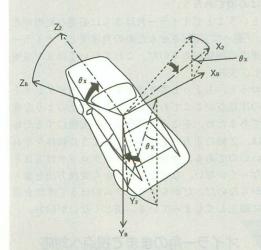
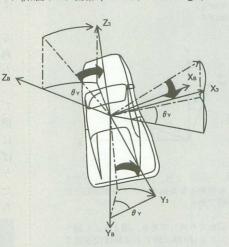
$$Rz=\begin{pmatrix}\cos\theta z & -\sin\theta z & 0\\ \sin\theta z & \cos\theta z & 0\\ 0 & 0 & 1\end{pmatrix}$$

合成

$$R=Ry\cdot Rx\cdot Rz=\begin{pmatrix}cYcZ+sXsYsZ & sXsYcZ-cYsZ & cXsY\\ cXsZ & cXcZ & -sX\\ sXcYsZ-sYcZ & sYsZ+sXcYcZ & cXcY\end{pmatrix}$$

cos=c，sin=s，θx=X，θy=Y，θz=Zとした

図3

(a)Z_B軸回りにθz回転（バンク角：bank angle）

$Y_BY_1X_BX_1$は同一平面上
すでにこの段階で物体の座標系と回転の座標系は違うものになる

(b)X_B軸回りにθx回転（仰角：elevation angle）

(c)Y_B軸回りにθy回転（方位角：azimuth angle）

X，Y，Zの回転行列とは，X_B，Y_B，Z_B軸に対する回転の行列である。決してXn，Yn，Zn(nは1〜3，図参照）に対する回転ではないのである。

一般的なオイラー角は，そのどれかをXn，Yn，Znに対する回転角として定義しているのである。そのままでは変換はかなり厳しい計算をしなければならないので，SLASHでは以上のようにオイラー角を定義する。

任意の姿勢，図では$X_3Y_3Z_3$の状態（物体の座標軸）に対する回転（オイラー角の定義に等しい回転である）を行うには，初期状態の軸に対する回転ができるよう，回転角をうまく計算しなければならない。このことについては今回は扱わない。

介した坪井氏による三角関数の合成を行えばより高速になるのである。このときに注意すべき点は，三角関数の足し算が出てくることである。最悪の場合，$4000＋$4000，つまり$8000でオーバーフローを起こしてしまうため，テーブルの適当な調整が必要となる。これは範囲を$3FFF～-$3FFFにするだけで十分。これで出てくる誤差は問題にならない。

SLASHとオイラー角

さて，このオイラー角であるが，実に扱いづらいものである。サンプルのテスタロッサを回してみた人なら体験していると思うが，任意の姿勢をさせるにはかなりの試行錯誤が必要だ。原因は先ほどのオイラー角の図で，ある程度見当がつくであろう。

どうしてこのようになってしまうのかは簡単である。3軸に対して同時に回転ができないからである。となれば，変換には順番がある，ということになる。SLASHでは，

変換行列　$R=R_Z\times R_X\times R_Y$

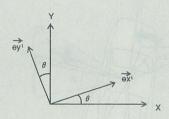
となっている。つまりZ軸まわり，X軸まわり，Y軸まわりの順番で座標を回転させる。

ここで，先ほどいった座標軸は回転しないことがはっきりする。SLASHの例であれば，まずZ軸まわり（画面奥）に回転させる。次にその変換された座標をX軸まわり（画面水平）に回転させ，その結果をY軸（画面垂直）に対して回転させているのである。X軸，Y軸は決して最初のZ軸の回転で影響を受けないので，X，Y軸の回転のときには物体の座標系とは違う軸での回転になってしまうのである。これは図3(a)のことである。

図4

2次元平面での回転行列

$$R=\begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}$$

である

第1行の（cosθ，-sinθ）とは，図の$\vec{ex'}$そのものである。つまりは，座標軸となる基本ベクトルが回転行列を形成しているのである

これは3次元の回転でも同じである。図2をよく調べてみてほしい，たとえばX軸に対する回転はYZ平面での回転であり，回転がX軸方向の基本ベクトルには，一切影響を与えていないのである。

次にこれはオイラー角ではない，といったことについて考えよう。一般的なオイラー角は，方位角，仰角，バンク角の順序である。このとき，方位角は初期のY軸に対する角度である。だがその次の仰角は変換後のX軸に対する角度，バンク角はさらに変換されたZ軸に対する角度を意味している（本当は，航空力学においての方位角はZ軸，仰角はY軸，バンク角はX軸となっているが，ここではSLASHの定義に従って表記している）。

しかし，行列による変換はすべて初期状態の軸による変換なのである。よってオイラー角の定義に従った回転を行うためには，それぞれの軸が初期状態の軸に一致している状態で回転させなければならない。ということで説明した変換は一般的なオイラー角とは順序が逆になっているのである。そして回転は常に初期状態の軸に対しての変換となっているのである。もし航空力学などの本があればそのオイラー角のページと比べてみるといい。そしてどの軸に対する回転角をオイラー角というのか，じっくりと照らし合わせてみてもらいたい。

以上より，SLASHにおいてオイラー角とは，図3のX_B，Y_B，Z_Bを基準とした回転角であり，角速度はX_3，Y_3，Z_3を基準とした回転角となっている。任意の姿勢からの定義どおりの回転のためには，このオイラー角速度と角速度との関係を導き出さねばならない。これがオイラー角による回転制御の基本である。

これとは別に12月号で丹氏が解説した基底ベクトルというものがあるが，物体を自由に制御する，ということを解決しているわけではない。物体を自由に回転させるためにはこの基底ベクトルを「狙って」回転させる必要があるのだ。結局はどのような表現をとっても，3軸に対する回転をうまく制御することは必須であろう。

ということでオイラー角はさらに必要な処理がある。「狙って」回転させるための角速度と，オイラー角速度の関係があるのだ。これについてはまだ説明なし。失礼。

おっと，ちょっと寄り道をするが，モデリング上での注意点がここでわかると思う。以上のような変換であるため，そこそこ都合のいい変換にするためには，Z軸の正方向を正面とするように物体を作ればいいのである。サンプルのテスタロッサは実はそうなっていない。なぜなら途中から変換方法を変えたからなのだ。だからプログラムのほうで座標を適当に細工してしまっている。美しくないが勘弁。

オイラー角のままで視点へ対応

物体を配置したら，次はワールド座標系の任意の

位置から任意の方向を見れなければ由緒正しき3Dシステムとはいえない。申しわけないことに，SLASH Ver1.0ではここまでは対応していなかった。というのも，当時の段階ではまだ座標系についてはあまり考えておらず，まあ基本部分があればあとはなんとかなるだろう，と甘く考えていたからだ。次のバージョンからは座標系もそこそこ強くすることはすでに発表してあるから，もうしばらく辛抱してもらいたい。

さて，まず物体の扱いを抜いて軽く考えよう。空間のオブジェクトが自由な回転さえしなければ，空間すべてが1つのオブジェクトとして扱える。このとき前述のようなオイラー角の定義に従えば，比較的都合のよさそうな変換はZ，Y，X（X軸→Y軸→Z軸）である。これだけでもある程度は空間内を自由に扱えるようになっている。さらに物体を加えるとすれば，物体の回転行列とこの空間の回転行列を合成することで対処できる。あるいは物体の回転行列によって変換した座標，これらはすでにワールド座標系の座標であり，これをさらに空間の回転行列で回転させてもかまわない。ただし，双方ともあまり効率のよい変換でないことはわかっているだろう。

本来視点のためにはこのオイラー角に頼っていてはいけないのだが，いずれ応用としてマップシステムを考えるときには比較的楽な方法となる。この方法については，改めて必要になったときに深く考えてみたいので，今回は単純な方法を考えていく。

変換とは

今度は座標系の基本となっている，行列による変換自体について考えたい。まず簡単に2次元平面での回転を思い出そう。しっかり覚えているであろうか。大学生である私にとっては当然過去の遺物であり，出くわすたびに本をチェックしなければならないのは泣けてくる。

さて，2次元平面の行列は2×2行列であった。平行移動は無視することにしよう。さて，賢明な読者ならば記憶に頼らずにこの行列を求めることができるはずである。第1行のパラメータを（a，b）とし，変換前の点を（x，y）とすれば，変換後のX＝ax＋byである。ではこのときの行列の各パラメータはなにを意味しているのであろうか。考えればなんとなくわかるであろうが，当然基本ベクトルを意味しているのである。もしかしたら当然このように習うかもしれないが，私はしっかりと記憶から消していた。図4を参照しよう。回転とはつまり基本ベクトルの置き換えそのものなのである。

ではこれを3次元に応用して考えてみてほしい。やはりこれも同じことなのである。試しにX軸まわりの回転でもやってみてほしい。X＝x＋0＋0で，Xは影響を受けず，あとのY及びZはY，Z平面上での回転となっていることがチェックできるだろう。

座標変換とは，結局は基本ベクトルの操作が問題なのである。本来スケーリングなども簡単に行えるのではあるが，それはアセンブラの都合により無視しているのである。また，特殊な効果として異常な変形，モーフィングとまではとてもいかないがつぶしたり伸ばしたりなどの気持ちいい変形も意のままに可能なのである。

視野変換

それでは任意の視線に対応できる変換を考えてみよう。申しわけないがSLASH Ver1.0にはまだない機能である。やろうと思えば可能なので，次バージョンにてサポートする。

座標変換の方法はオイラー角の定義に従う。回転順序は，X，Y，Z軸の順番で都合がいいことは前述してある。まずZについてであるが，バンクは最終的な画角の調整以外にはいっさい関係のないパラメータであるため，変換アルゴリズムからは外して考えるべきである。あくまでも「最終的な」であるからZ軸による変換は最後に行う。残りのX，Y軸の順番であるが，考えやすいモデルはY軸が地面に垂直なものである。となると，制御のために重要なのは仰角よりも方位角であろう。よってX，Y軸はこのままの順序がやりやすい。ということでこの順序で回転させることによって視野変換を行うことにしよう。

変換手順は次のとおりである。

I：ワールド座標系原点を視点に移動
II：視野X軸，Y軸まわりに回転
III：視野Z軸まわりに回転させて画角を決定

ではこの回転はどのように決定できるのであろうか。視線を表すパラメータとして必要なのは視点，注視点である。これは当然ワールド座標系内の任意の座標である。ここから回転角を求めればよい。しかし回転行列を構成するものは三角関数であった。よって直接その値が求められれば角度である必要がなく，素直に行列を求めることができる。

ではすべてを解説する図5を見てもらいたい。結局は任意のベクトルから三角関数を求めるルーチンがあれば，あとはその値を直接3軸それぞれの回転行列に当てるだけで解決する。問題点はまず範囲が絞られることである。仰角は−90°～90°の範囲に絞られてしまう。まあまあり問題でもないか。この方法であれば仕方のないことであろう。もうひとつは平方根が出てしまうことである。アセンブラとしてはやや苦しい処理であるが，ニュートン法があれば平方根もそれほど痛くはない。

次のステップ

座標系に関して初歩的なことを綴ってみた。だがやはりこれができても自由に空間は制御できないのは確かである。まずこのオイラー角の定義と行列の意味，つまりは基底ベクトルによる変換との差をしっかりと体に叩き込んでおきたい。それらをさらに操るのは次のステップの問題である。

一応これだけでも適当な3Dモノは作れるが，正しい3Dモノを制御するまでいたらないのである。もし自由に空間を飛び回りたいなら，無用な物体，地上を省くためのマップ制御システムも必要だし，任意の姿勢から任意の回転を行うためには，さらにオイラー角の発展をさせなければならない。いずれも回を追って解説することになるであろう。ということで来月は透視投影，そしてポリゴンのZ方向クリッピング，座標変換のいろいろな都合について考えてみたいと思っている。

今後のバージョン

さて，またもやバージョンアップについてであるが，グラフィック疑似スプライトなんかも導入することになった。拡大縮小専用と回転つきのやつが新たに加わるはず。なんだかこのままいくとマッピングだとかレイトレーサーだとかもやりたくなってしまうが，それはそれってことにしよう。そういえば最近やたらと忙しいな。すべて学校が悪いのだろう。だいたいいまの時代に車での登校禁止なんて非文明的だよ。結局は忙しくて作業の時間が苦しいってことだ。でも野望だけは捨てられないから，どちらかというと学校を捨てるくらいの気持ちでがんばろうと思うからよろしく，ってことでまた来月。

図5

(a)原点を視点に移す

(b-1)まずY軸に対してαを回転させる

$$\cos\alpha=\frac{X_H-X_V}{\sqrt{(X_H-X_V)^2+(Z_H-Z_V)^2}}$$

$$\sin\alpha=\frac{Z_H-Z_V}{\sqrt{(X_H-X_V)^2+(Z_H-Z_V)^2}}$$

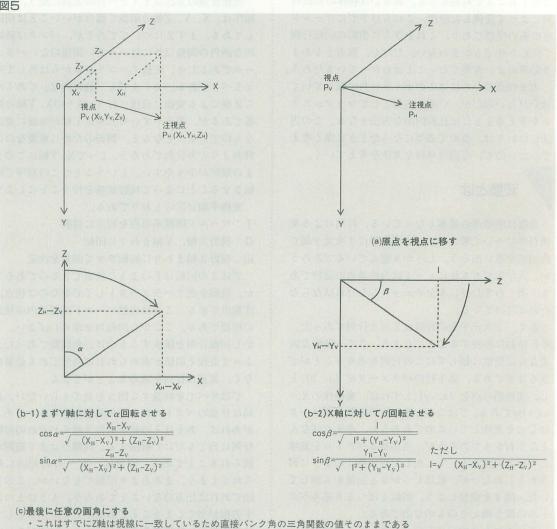
(b-2)X軸に対してβ回転させる

$$\cos\beta=\frac{l}{\sqrt{l^2+(Y_H-Y_V)^2}}$$

$$\sin\beta=\frac{Y_H-Y_V}{\sqrt{l^2+(Y_H-Y_V)^2}}$$

ただし $l=\sqrt{(X_H-X_V)^2+(Z_H-Z_V)^2}$

(c)最後に任意の画角にする
・これはすでにZ軸は視線に一致しているため直接バンク角の三角関数の値そのままである
・あとはこれらを図2の行列Rに代入すれば変換行列が求められる

THE SENTINEL

〈対応機種一覧〉●MZ-80 K/C/700/1500●MZ-80 B/2000●MZ-2500/2861●X1●X1 turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA/7●FM-7/77/AV●MSX/2/2+/turbo R●PC-286/386/486/9801/98/9821●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。

第139部　S-OSで学ぶZ80マシン語講座(2)

●「YGCS ver.0.20」仕様公開延期！

まずはごめんなさい。ページの都合上,「YGCS ver.0.20」の仕様公開が来月号に延期されてしまいました。

そこで，せっかく使えるものが送られてきたのにそれを公開するスペースがないという，情けない理由でいたずらに時間を浪費するのはもったいない。今回は特別に全ソースリストとオブジェクトのモニタ募集を30名行します。もちろん，モニタということですから「YGCS ver.0.20」を実際に使用して，レポートを書いてもらうことになります。

配布から1カ月後までにシステムを使用した感想，作ったプログラムなどを返送していただきます。必ずなにか具体的なものを作ってくれるのが望ましいのですが，なにぶん短い期間であることですし，アイデアだけでもOKです。

なお，応募資格は，S-OSシステム所有者で，Z80アセンブラを使えること。そして，このモニタレポートの目的は，最終的な仕様の決定です。リストの最適化よりも，具体的にどのようなサービスコールを用意したらいいかという意見をバシバシお書きください。

配布メディアは5インチ2D，X68000，PC-9801用5インチ2HD，そして可能であれば3.5インチ2DD（MSX用）を用意します。

官製ハガキに住所，氏名，年齢，職業，電話番号，使用機種名，アセンブラ歴を明記のうえ,

Oh!X編集部
S-OS「YGCS ver.0.20」モニタ希望係

までお送りください。

発送作業は，来年1月から行う予定なので気長にお待ちください。

●謎の「CueSystem」

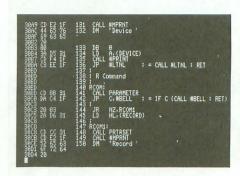
今月は「これからのTHE SENTINEL」をお休みにして，以前からたびたび投稿のあった「CueSystem」を紹介しましょう。

これは，S-OS"SWORD"でX1の機能をフル活用するため開発された，SLANG用ライブラリを中心としたシステムです。X1というマシンに徹底的にこだわり，X1のもつポテンシャルを最大限に引き出すために作られた，非常に根性の入ったライブラリ集といえます。

今回，バージョン1.00として完成し，販売を行うということで，急遽紹介することになりました。

本来ならば,「USERS WORKS」で紹介するのが筋でしょうが，S-OSを使用しているということもあり，この「THE SENTINEL」のコーナーで取り扱うという，異例の事態となりました。

では，軽く概要を説明します。本システムは,

■SLANG用X1シリーズ専用ライブラリ「Quetzal」

X1 turboZのアナログモードまで，ほぼすべてのハードウェア機能をサポートしたライブラリ。

■フルスクリーンキャラクタエディタ「GRID」

グラフィックエディタとキャラクタエディタをドッキングさせた，320×200ドット専用フルスクリーンキャラクタエディタ。ライブラリ「Quetzal」を実際に使用したサンプルです。

以上を中心に構成されており，価格は,

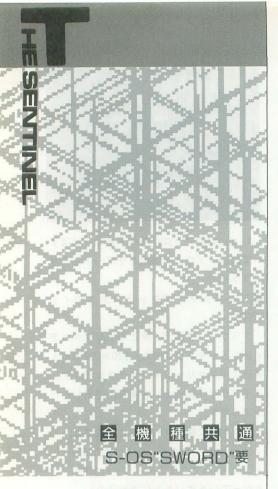
・CueSystemプログラム一式　4,000円
・CueSystemマニュアル　　　1,000円

となっています（ともに送料込み)。

入手方法は，欲しい商品名と住所，氏名，電話番号を書いたメモと，料金分の郵便小為替か現金書留を下記の住所にお送りください。

〒988　宮城県気仙沼市字松川37-1
三浦 光雄

X1をもっと使いこなしたい人，システムに興味のある人は，問い合わせをしてみてはいかがでしょうか？

1993■インデックス

S-OSで学ぶ Z80マシン語講座(2)

Itou Masahiko
伊藤 雅彦

今回は，11月号で紹介したリストをもとに，どのように命令が組み合わされてプログラムが形成されているか，じっくり解説していきます。がんばってリストを読みこなしてください。

いよいよ，今月から本題に入ります。対象とする読者としては，「個々の命令は本を読んだりしてわかったような気がするけど，これでどうやってプログラムを作るんだろう」という人を想定しています。ですから，本文ではマシン語自体にはあまり立ち入らないで，テンポよく進んでいきたいと思っています。

でも，マシン語に関心をもっていなかった人にも，「お，これは面白いかもしんない」と興味をもってもらいたいので，適当に囲みを作って基礎的なことについて若干の補足説明を加えます。本文でわからないところがあったら参照してみてください。ただし，本腰入れてマシン語に挑戦してみたいと思ったら，ちゃんとした入門書を手に入れましょう。基礎知識を整理して覚えられると思いますよ。

プログラム構造

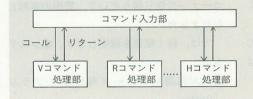
さて，ディスクエディタADDIEを作りながらマシン語の解説をするということですから，さっそく作り始めましょうか。

まず，この手のユーティリティの形には，黄金のパターンがあるわけです。DREAM, TRADE, SLANG, REDA, S-OS“SWORD”のモニタなど。これらはすべて，

1)　プロンプト（コマンドを入力しようとする行の頭に出る“#”など）を画面に出して，コマンドの入力を受けつける
2)　入力されたコマンドに従った処理を実行
3)　1)に戻る

ということを繰り返しています。どんな処理ができるかによって，ディスクエディタになったり，デバッガになったり，コンパイラになったりするんですね。

今回のADDIEもこのパターンに倣います。ですからプログラムは大まかにいって，コマンド入力部と，各コマンドそれぞれの処理部に分けられます。各コマンドの処理部はコマンド入力部のサブルーチンとして位置づけ，入力されたコマンドによって，該当するコマンド処理部が呼び出されるという構造にします（図1）。

図1　プログラム構造

ワークエリアの初期化

概略の見当をつけたところで，メインルーチンを作ってみましょう。メインルーチンの主な処理は，コマンド入力処理ともうひとつ，起動時のワークエリア（変数を記憶しておくメモリ領域）の初期化があります。これさえ出来上がれば，とりあえずコマンド入力を受けつけてくれるプログラムを動かすことができます。そして各コマンド処理のサブルーチンをつけ加えていけば，使えるコマンドが増えていくのです。

まず，ワークの初期化部分を考えていきます。とはいうものの，どんなワークを用意するかを決めないと，初期化もなにもありませんね。リスト1を見てください。ここにワークの定義が書いてあります。ざっと考えて，これは必要だなと思ったのがこれらのワークです。

これらのなかで初期化する必要のあるワークは，WIDMODE, DEVICE, RECORD, WRCBK, PRTRSWの5つです。以下，ワークの役割を解説します。

・WIDMODE：RコマンドやFコマンドで画面モードによって表示を変えるため，画面モードを知るためのワークです。初期化時の画面モードによって，0か1を設定しておきます。

・DEVICE：各種コマンドでデバイスを読み書きするときにどのデバイスにアクセスするのかを指定しなくてはいけません。そのときに必要なデバイス名を保持するワークです。初期設定値としてデフォルトデバイスを設定しておくのがいいでしょう。

・RECORD，WRCBK：R，Wコマンドでパラメータ省略時にレコード番号を決めるためのもの。0で初期化します。

・PRTRSW：プリンタ出力モードを保持するワーク。最初は印字OFF状態にしておきたいので，0を設定します。

さて，このワーク初期化の処理をアセンブラで書くとどうなるかというと，リスト2の41～64行のようになります。上から順に見ていきましょう。

最初にMAINというラベルが定義されています。これはどこからも参照されない，いってみれば無駄なラベルです。ただ，メインルーチンの最初に“MAIN:”と書くと，「ここがプログラムの実行開始アドレスなんだぞ」と主張しているような感じがするので，私は必ず書いています。

次の行からWIDMODEの初期化

をやっています。WIDMODEの初期化のためには画面モードを知る必要がありますが，このためにS-OSのシステムワークエリアの#WIDTHを参照しています。

S-OS内にはシステムに関する情報が書き込まれたワークエリアがあって，ユーザーはそれを自由に参照できます（参照が禁止されているものも一部あり）。場合によっては書き込みを行うこともあります。このワークの一覧表は，1993年7月号など，S-OS"SWORD"（どの機種のものでも）が掲載された号に，ほとんど載っていますから見ておいてください。

#WIDTHは1バイトのワークで，画面モードが40桁なら40(28_H)，80桁なら80(50_H)が入っています。ですから，ここが40ならWIDMODEに0を設定して，そうでなければ1を設定するというプログラムを作れば，WIDMODEの初期化ができます。それをやっているのがリスト2の42～47行です。1行ごとに言葉で説明すると，

42：Aに#WIDTHの値を入れる
43：40かどうかチェック
44：40ならMAIN1へジャンプ
45：Aに初期値1を入れる
46：MAIN1（ラベル）
47：WIDMODEに初期値を設定

となります。目で追って理解してみてください。Aレジスタの値とZフラグの状態に注目して，#WIDTHが40の場合，そして80の場合どうなるか，という具合です。

ここで，#WIDTHの値が40かどうかの判断にCP命令ではなく，SUB命令を使っています。もちろん，これにはちゃんとした理由があります。それは，もし#WIDTHの値が40なら，Aレジスタが自動的にWIDMODEへの設定値となる0になる，という利点があるのです。47行でAレジスタの値を初期値として設定していますから，ここへジャンプするだけでよくなるんですね。CP命令で同じことをるとどんなプログラムになるかを考えてみれば，SUB命令にした意味がよくわかるでしょう。

もっとも，比較のときには，まずCP命令を使う，という基本は忘れないでください。初めからこの手の小技を狙いすぎると，どういうプログラムを組んだらいいのか，かえって考えがまとまらなくなりますから。

では，次にDEVICEの初期化をしましょう。DEVICEにはデフォルトデバイスを設定しますが，デフォルトデバイスを知るにはシステムサブルーチンを使う必要があります（システムサブルーチンの一覧表もS-OS"SWORD"が掲載されている号に載っています)。使い方は普通のサブルーチンと同じで，必要な引数をレジスタにセットして，CALL命令でコールするだけです。で，デフォルトデバイスを知るためのサブルーチンは#RDVSWです。ここをコールするとデフォルトデバイスのキャラクタコードがAレジスタに入れられて戻ってきます。

ということは，#RDVSWをコールして，LD(DEVICE)，　Aとすれば処理が完了しそうですが，ちょっと待ってください。ADDIEで扱うデバイスはA～DのフロッピーディスクとEのRAMディスクだけです。一方，デフォルトデバイスはQやSやTの可能性もあるわけです。それがDEVICEに設定されてはまずいですから，もしデフォルトデバイスがA～E以外だったら，デバイスAを初期値として設定することにします。

リスト2を見てみましょう。DEVICEの初期化処理は49～57行でやっています。各行を言葉で説明すると，

49：Aにデフォルトデバイスを入れる
50：A(41_H)より小さいかチェック
51：小さければMAIN2へジャンプ
52：E(45_H)より大きいかチェック
53：大きくなければMAIN3へジャンプ
54：MAIN2（ラベル）
55：Aに初期値A(41_H)を入れる
56：MAIN3（ラベル）
57：DEVICEに初期値を設定

となります。CP命令を2回使ってキャラクタコードが，A～Eかどうかをチェックしているわけです。Aレジスタの値とCyフラグの状態に注意しながら目で追って，デフォルトデバイスがA～Eでなかった場合にだけ55行のLD命令を通ることを確認してください。あ，デフォルトデバイスのキャラクタコードが，Aより小さいということは実際にありえないので，50，51行目は無駄だということに，いま気がつきました。

でも，不測の事態に備えて（どんな事態だか)，どんなことがあってもDEVICEにはA～E以外のキャラクタは入れさせないのだという意味で，修正はしません。

59～64行では，PRTRSW, WRCBK, RECORDを0で初期化しています。ここは特

リスト1

```
31D4           391 ;
31D4           392 ; Work Area
31D4           393 ;
31D4           394 WIDMODE:       ; 表示桁80桁(1)／40桁(0)
31D4 00        395  DS 1
31D5           396 DEVICE:        ; デバイス名
31D5 00        397  DS 1
31D6           398 RECORD:        ; R・Wコマンドのレコード省略時
値
31D6 00 00     399  DS 2
31D8           400 WRCBK:         ; Wコマンドで省略時値を1戻す(1
)／戻さない(0)
31D8 00        401  DS 1
31D9           402 KBPTR:         ; キーバッファの注目アドレス
31D9 00 00     403  DS 2
31DB           404 PRTRSW:        ; 印字ON(1)／OFF(0)
31DB 00        405  DS 1
```

リスト2

```
3000            38 ;
3000            39 ; Main
3000            40 ;
3000            41 MAIN:
3000 3A 5C 1F   42  LD   A,(#WIDTH)
3003 D6 28      43  SUB  40
3005 28 02      44  JR   Z,MAIN1
3007 3E 01      45  LD   A,1
3009            46 MAIN1:
3009 32 D4 31   47  LD   (WIDMODE),A
300C            48 ;
300C CD 24 20   49  CALL #RDVSW
300F FE 41      50  CP   'A'
3011 38 04      51  JR   C,MAIN2
3013 FE 46      52  CP   'E'+1
3015 38 02      53  JR   C,MAIN3
3017            54 MAIN2:
3017 3E 41      55  LD   A,'A'
3019            56 MAIN3:
3019 32 D5 31   57  LD   (DEVICE),A
301C            58 ;
301C AF         59  XOR  A          ; A=0
301D 32 DB 31   60  LD   (PRTRSW),A
3020 32 D8 31   61  LD   (WRCBK),A
3023 6F         62  LD   L,A
3024 67         63  LD   H,A
3025 22 D6 31   64  LD   (RECORD),HL
```

今月使用した システムサブルーチン&ワークエリア

●サブルーチン

#RDVSW：デフォルトデバイスをAレジスタに入れる

#LPTOF：画面表示時に同時にプリンタ出力をしないようにする

#PRINT：カーソル位置にAレジスタの文字を表示

#GETL：1行入力を行い，行データをDEレジスタペアのアドレスから256バイトの領域に格納。カーソルは入力行の次の行の先頭に移動

#MPRNT：CALL命令の次のアドレスから00_Hがあるまでカーソル位置より文字列表示

#LTNL：改行

#LPTON：画面表示時にプリンタにも出力するようにする

●ワークエリア

#WIDTH：現在の画面モード（40桁なら40，80桁なら80)

#KBFAD：キー入力バッファの先頭アドレス

に説明する必要はないでしょう。ただ，最初の，

XOR　A

はちょっとわかりにくいかもしれません。これはAレジスタの値を0にするためのものです。なぜこれでAレジスタの値が0になるのかわかりますか？　AレジスタとAレジスタの排他的論理和をとる，つまり同じ値の排他的論理和なんだから……ね，Aレジスタの元の値がなんだったとしても，結果が0になるでしょ。あと，

SUB　A

としても同じことです。どうしてそうなるかはこっちのほうがわかりやすいかな。これらは，同じ処理をするために，

LD　A,0

とするより命令が短く，高速であるためよく使われます。覚えておきましょう。ただし，LD命令と違ってフラグ変化があることに注意してください。

コマンド入力処理

さあ，今度はコマンド入力の部分を作ります。コマンド入力処理の概要は，プロンプトを表示してコマンド入力を受けつけ，入力されたコマンドによって各コマンド処理ルーチンをコールして，またプロンプトを表示して……というループ処理です。もう少し細かく手順を考えてみると，

1)　プリンタ出力OFF
2)　プロンプトを表示
3)　1行入力
4)　入力された行の頭にプロンプトがあるかどうかチェック
5)　入力コマンドによりコマンド処理ルーチンをコール
6)　1)に戻る

ということになりそうです。最初にプリンタ出力をOFFにするのは，各コマンド処理ルーチンをコールして戻ってきたときにプリンタ出力がONの状態になっていることがあるためで，ここでOFF状態にリセットしています。

では，実際にどういうプログラムになるか，リスト3を見ていきましょう。66〜110行がコマンド入力処理の部分です。

最初にするのはプリンタ出力のOFFでしたね。これは67行のCALL命令ひとつで終わっています。次のプロンプト表示も68，69行を見ていただければわかるとおり，#PRINTをコールするだけです。これではカーソル位置の設定をなにもしていないので，プロンプトが行頭に表示される保証がないんじゃないかと思われるかもしれませんが，「各コマンド処理ルーチンは必ずカーソルを行頭に移動させてからリターンする」という約束ごとを作っておけば大丈夫です。

次にしなくてはいけない1行入力も，#GETLルーチンをコールするだけ。完全に自前のプログラムでやろうとすると，カーソル点滅やらキー入力やら画面へのエコーバックやらの処理でとてもうっとうしいことになります。ところが，#GETLルーチンはそういった処理を全部引き受けてくれ，さらにDEレジスタペアのアドレスから256バイトの領域に入力行の文字列データを格納してリターンしてくれるんです。

そして，その256バイトのキー入力バッファもユーザーのメモリ領域から用意する必要はなく，システムワーク#KBFADに書かれたアドレスから256バイトを使うことができます。リスト3では70，71行が#GETLをコールしている部分です。次に，72〜74行で行頭にプロンプトがあるかどうかをチェックして，なければ行入力は無効としてMAIN4へ戻っています。

76行からが入力されたコマンドの処理ルーチンをコールする部分になります。まず最初にワークKBPTRにキー入力バッファの先頭アドレス＋1を入れて初期化しています。入力コマンドの内容を解析していく際には，キー入力バッファを先頭から後ろのほうへ順に見て解析を進めていくのですが，KBPTRはまだ解析の終わっていない先頭のアドレスを入れておくワークです。ここで，行の先頭はプロンプトであるという解析が終わっているので，KBPTRの初期値として先頭から2文字目のアドレスを入れているわけです。

続いてSPCUT，CAPITALと順にコールしています。SPCUTはキー入力バッファの注目点(KBPTRが指しているところ)からスペースを飛ばして，次にある文字をAレジスタに入れるサブルーチンです(図2)。KBPTRの値はAレジスタに入れた文字＋1を指すように更新されます。CAPITALはAレジスタに入っている文字が，アルファベットの小文字なら大文字に変換します。どちらもとても短いルーチンですので，自力で解読してみてください。

この2つのサブルーチンを通してAレジスタに得られたキャラクタが“V”なのか

リスト3

```
3028            66 MAIN4:
3028 CD D6 1F   67  CALL #LPTOF
302B 3E 3D      68  LD   A,PROMPT
302D CD F4 1F   69  CALL #PRINT
3030 ED 5B 76   70  LD   DE,(#KBFAD)
3033 1F
3034 CD D3 1F   71  CALL #GETL
3037 1A         72  LD   A,(DE)
3038 FE 3D      73  CP   PROMPT
303A 20 EC      74  JR   NZ,MAIN4
303C            75 ;
303C 13         76  INC  DE
303D ED 53 D9   77  LD   (KBPTR),DE
3040 31
3041 CD B6 31   78  CALL SPCUT
3044 CD C3 31   79  CALL CAPITAL
3047 21 28 30   80  LD   HL,MAIN4
304A E5         81  PUSH HL         ; RET で MAIN4 にジャンプさ
せる細工
304B FE 56      82  CP   'V'
304D CA 8C 30   83  JP   Z,VCOM
3050 FE 52      84  CP   'R'
3052 CA BD 30   85  JP   Z,RCOM
3055 FE 57      86  CP   'W'
3057 CA 3E 31   87. JP   Z,WCOM
305A FE 46      88  CP   'F'
305C CA 3F 31   89  JP   Z,FCOM
305F FE 44      90  CP   'D'
3061 CA 40 31   91  JP   Z,DCOM
3064 FE 43      92  CP   'C'
3066 CA 41 31   93  JP   Z,CCOM
3069 FE 4F      94  CP   'O'
306B CA 42 31   95  JP   Z,OCOM
306E FE 4B      96  CP   'K'
3070 CA 43 31   97  JP   Z,KCOM
3073 FE 53      98  CP   'S'
3075 CA 44 31   99  JP   Z,SCOM
3078 FE 54     100  CP   'T'
307A CA 45 31  101  JP   Z,TCOM
307D FE 48     102  CP   'H'
307F CA 46 31  103  JP   Z,HCOM
3082 FE 50     104  CP   'P'
3084 CA 47 31  105  JP   Z,PCOM
3087 FE 51     106  CP   'Q'
3089 C0        107  RET  NZ
308A           108 ;
308A E1        109  POP  HL         ; 細工を解除
308B C9        110  RET
```

図2　SPCUTルーチンの機能

"R"なのかなどによって，どのコマンドが入力されたかを判断するわけです。SPCUTでスペースを飛ばす処理が入っていますから，コマンド入力時にプロンプトとコマンド文字の間を空けて，

```
  #  V
```

と入力しても受けつけられますし，またCAPITALルーチンで小文字を大文字に変換していますので，

```
  #v
```

のような小文字入力にも対応します。

さて，これでAレジスタにコマンド文字が入っているわけですから，あとは，

```
      CP     "V"
      JR     NZ,MAIN5
      CALL   VCOM
      JR     MAIN4
  MAIN5:
      CP     "R"
         ⋮
```

というように，各コマンド処理ルーチンをコールする処理を並べていけばいいですね。82行以降を見てください。ここではCALL命令の代わりにJP命令でコマンド処理ルーチンを呼んでいます。プログラムが短いし，すっきりしているでしょう？

でも，これではコマンド処理ルーチンからリターンするときにおかしくなってしまうのでは，とお思いの方，その通りです。そこで80，81行の怪しげな細工が必要になってきます。ここではMAIN4のアドレスをスタックにPUSHしています。こうすると，コマンド処理ルーチンからリターンしたあと，MAIN4に処理が移ります。ここのところはスタックとサブルーチンコールの仕組みをしっかり理解していないとわからないでしょう。ま，すぐにはわからなくても，そのうちわかってもらえればいいです。前回のときにいったでしょう，初心者向けのプログラムを組む気はないって。

Aレジスタに入っているキャラクタがコマンドの文字ではなかった場合に，107行のRET命令でMAIN4にジャンプします。なぜそうなるのかといえば，それも例の怪しげな細工のおかげです。もしも，Aレジスタが"Q"だったら，つまり終了コマンドが入力されたら109行に処理が移ります。ここでは，MAIN4の値をスタックにPUSHするという例の細工を，POP命令を実行することによって解除してから，RET命令でプログラムの実行を終了しています。

S-OS上のアプリケーションプログラムは，S-OSシステムからサブルーチンコールされるという形で実行されますので，RET命令でプログラムの実行を終えてS-OSシステムに戻ることができるのです。S-OS "SWORD" でアプリケーションを作るための基本事項ですので覚えておきましょう。

Vコマンド

メイン処理ができたら，あとはコマンド処理をばんばんつけていくだけです。まずはVコマンドからいきましょう。

Vコマンドはデバイスの変更，表示です。パラメータとしてA～Eが指定された場合はデバイスの変更と，変更後のデバイスの表示を行い，それ以外の場合は現在のデバイスを表示することにします。この手順を整理すると，

1) パラメータを取得
2) パラメータがA～Eなら，それをワークDEVICEに設定
3) DEVICEの内容を表示

となります。

リスト4，114行からのVコマンドルーチンを見ていきましょう。まずパラメータの取得ですが，このルーチンがコールされた時点で，KBPTRはコマンド文字の"V"+1を指しているため，SPCUTをコールするだけでよいとわかります。これでパラメータをAレジスタに取り出せます。そして小文字→大文字変換をしてからパラメータがA～Eかどうか判断して，A～EでなければVCOM1へジャンプします。

122～127行はパラメータがA～Eだった場合だけ実行される処理になります。DEVICEにパラメータ値を入れてデバイスの変更をしています。それからRECORD，WRCBKを0にクリアしています。これらのワークはR，Wコマンドのパラメータ省略時値に関するもので，デバイスが変わったらリセットしています。

ラベルVCOM1以降はDEVICEの内容表示です。具体的には，

```
  DeviceA
```

というような形で表示します。

ここでは，まずPRTRSETをコールしていますが，このサブルーチンはプリンタ出力モードがONなら#LPTONをコールするというものです。ここのサブルーチン内にある，

```
  OR  A
```

という命令は要チェックです。この命令の意味は，「Aレジスタの値とAレジスタの値の論理和(OR)をとり，結果をAレジスタに入れる」ということです。Aレジスタの値どうしの論理和をとってもAレジスタの値はなにも変わりません。そう，この命令は一見なんの意味もなさそうです。

でも，フラグレジスタの変化に注目してください。OR命令は演算結果が0ならZフラグを1にし，0以外ならば0にします。ということは，"OR A"を実行すると，Aの値が0ならZフラグが1になり，そうでなければ0になります。これでAの値が0かどうかのチェックができるわけです。

それだったら比較命令で"CP 0"とやったほうが素直じゃないかと思うかもしれません。しかし，命令のバイト数が少なく，実行時間も短いのです。0チェックというのは結構頻繁にあるものですから，これも覚えておいてください。

ちなみに，"AND A"でも同じことができます。また，"OR A"でAレジスタの0チェックができるからといって，"OR B"でBレジスタの0チェックはできませんので，間違えないように。

Vコマンド処理の説明に戻りましょう。PRTRSETをコールして，プリンタ出力モードがONのときにはプリンタにも出力することにします。そして，"Device"という文字列を表示するために，#MPRNTルーチンを使っています。これは変な動作をするサブルーチンで，CALL命令の置かれた直後のアドレスからのデータを，0があるところまで文字表示して，0のデータの次のアドレスにリターンするものです。CALL命令の直後にリターンしないというところがなんだか奇妙でしょう。なんでこ

リスト4

```
308C            111 ;
308C            112 ; V Command
308C            113 ;
308C            114 VCOM:
308C CD B6 31   115  CALL SPCUT
308F CD C3 31   116  CALL CAPITAL
3092 FE 41      117  CP   'A'
3094 38 10      118  JR   C,VCOM1
3096 FE 46      119  CP   'E'+1
3098 30 0C      120  JR   NC,VCOM1
309A            121 ;
309A 32 D5 31   122  LD   (DEVICE),A
309D AF         123  XOR  A          ; A=0
309E 32 D8 31   124  LD   (WRCBK),A
30A1 6F         125  LD   L,A
30A2 67         126  LD   H,A
30A3 22 D6 31   127  LD   (RECORD),HL
30A6            128 ;
30A6            129 VCOM1:
30A6 CD CC 31   130  CALL PRTRSET
30A9 CD E2 1F   131  CALL #MPRNT
30AC 44 65 76   132  DM   'Device '
30AF 69 63 65
30B2 20
30B3 00         133  DB   0
30B4 3A D5 31   134  LD   A,(DEVICE)
30B7 CD F4 1F   135  CALL #PRINT
30BA C3 EE 1F   136  JP   #LTNL      ; = CALL #LTNL : RET
```

んなことができるのかというと，スタックをうまくいじってるからなんですよ，とだけいっておきます。各自考えてみましょう。

132，133行でDEF疑似命令を使っていますが，この意味は12月号のREDAの記事で理解してください。DEF疑似命令の仕様はREDAとZEDAでは細かい部分で異なっているんですが，このプログラムではどちらでも大丈夫な書き方をしています。

135行でデバイス表示を終えたあと，次のJP命令で#LTNLへジャンプしています。コマンド処理ルーチンからリターンするときにはカーソルが行頭になければならないという約束を，メインルーチン処理のところで作りましたね。だから最後に改行処理が必要になるんです。

ここで，サブルーチンにJP命令で飛んでいるのは変だなと思われるかもしれません。これは，

```
	CALL	#LTNL
	RET
```

をひとまとめにしたものと思ってください。CALL，RET命令の動作をよく考えてみれば，この意味も理解できることと思います。

今月はこれまでです。結局，メインルーチンとVコマンド処理しか説明できませんでした。次回からはコマンドを追加していきたいんですが，どうなるかは成り行きしだいだったりします。ではまた。

Z80基礎知識(1)

●レジスタ

CPUの中にはレジスタという変数みたいなものがあります。CPUというのはメモリ上のデータをいろいろといじくる装置なんですが，メモリにあるデータはレジスタに入れてからレジスタ上でいじくっています。Z80のレジスタ構成は図1のとおりです。

プログラムを作るうえで重要なのがA,B,C,D,E,H,Lのレジスタ。特にAレジスタはいろいろな演算ができるためよく使われます。どのレジスタでどんなことができるかは，命令表を眺めてつかんでおくといいでしょう。

また，2つのレジスタをつなげて使い，16ビットレジスタとして使うことができます。可能なのはBとC，DとE，HとLで，それぞれBCレジスタペア，DEレジスタペア，HLレジスタペアと呼びます。

Fはフラグレジスタで，主に演算関係の命令を実行したときに，演算結果によって値が変化するレジスタです。8ビットそれぞれが独立した意味をもっていて，重要なのがZ(ゼロ)ビットとCy(キャリ)ビットです。Zビットは演算結果が0になったときに1になり，それ以外なら0になります。Cyビットは演算で桁あふれか桁借りが起きたときに1になり，それ以外なら0になります。各命令によりフラグレジスタがどう変化するか，だいたいわかるようになったでしょう。

そのほかのレジスタの説明は省略。必要に応じて説明します。

●キャラクタコード

画面に表示される文字には，アルファベット，数字，カタカナなどいろいろありますが，これらの文字にはひとつひとつ番号が割り当てられています。“X”は58_H，“1”は31_Hというように。この番号をキャラクタコードといいます。メモリ上では普通このキャラクタコードで，文字データが記憶されています。S-OS“SWORD”では，表1のキャラクタコードを使います。

図1 (レジスタの項) レジスタ構成

A (8bit)	F S Z H P N C	A′	F′
B	C	B′	C′
D	E	D′	E′
H	L	H′	L′
I	R		
IX (16bit)		IY	
SP		PC	

表1 S-OSキャラクタコード

下位＼上位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	nul		SP	0	@	P		p					タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			“	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			S	4	D	T	d	t				エ	ト	ヤ		
5			%	5	E	U	e	u				オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B		BRK	+	;	K	〔	k	■			ォ	サ	ヒ	ロ		
C		→	,	<	L		l				ャ	シ	フ	ワ		
D	CLS	←	−	=	M	〕	m				ュ	ス	ヘ	ン		
E	CR	↑	.	>	N	^	n				ョ	セ	ホ	゛		
F		↓	／	?	O		o	π			ッ	ソ	マ	゜		

▶全機種共通システムインデックス◀

＊以下のアプリケーションは，基本システムであるS-OS“MACE”またはS-OS“SWORD”がないと動作しませんのでご注意ください。

プロはMacを使わない？

Mac雑誌は全部でいくつ？

業績悪化で会長は責任とってやめたとはいえ，世の中あきれかえるほどのマックマックの大行進。ソフトバンクもここに至ってMacintosh専門誌「MacUser」を出してきました。「MacUser」の先輩誌として僕がいま覚えているのは，「MACLIFE」「MACPOWER」「Macworld」「Mac Japan Bros.」「Mac Japan Active」「日経Mac」で，これで知っているだけでも全部で7冊というわけです。

これらの雑誌を全部買い揃えようとしたら，それこそ安いMacなら毎年1～2台買えてしまいます。ただ，必要な情報もそれなりに分散しているので，ちょいちょい気軽に買ってしまいます。そして，ふと気がつくとMac関係の雑誌でちょっとした山が形成されてしまっているのです（専門書や学会誌などの山に比べれば，ごくごく小さな山ですが）。

雑誌が山になった状態というのは，ちょうど検索手段のないデータベースのようなもので，ときどき，あの印象的な記事をもう一度読みたいとか，このソフトの使い方に関する情報はどこにあったっけとか，ずいぶん安いフルカラーのレーザープリンタ発売の記事はどの雑誌だっけとか思い立っても，なかなか5分以内に検索が終了しなくなってきました。

やはり，月刊誌でも索引があればいいですね，広告主の索引はありますが。半年に1回まとめるのでもがまんしますけれど。ただし，記事のタイトルを寄せ集めるというのではなくて，ちゃんとキーワードを記事のなかから拾い出して，それをあいうえお順ABC順にしてほしいのです。

その点，Oh!Xの「FILES Oh!X」というコーナーはそれなりに評価できます。ほかの雑誌を毎号毎号，機種ごとにチェックするということをずっと続けているのですからね。今度の新しい雑誌にもこれにならって，とりあえずソフトやハードの名前などで（ほかの雑誌の分もできるなら含めて）ピックアップして索引を作ってくれることを，実は期待していたのですが……。でも，実際問題として，相当な作業量になるのでしょう（しかし，それが多くの読者の作業量の削減につながるのです）。

山根氏の批判記事

ところで，その山のような記事のなかで最近最も印象に残ったのが，山根一眞氏の書いた記事（文献1）です。山根氏がMacworld誌に書いている連載，今回は「ビジネスマシンではなかったMac」というタイトルがつけられています。

1ページのこの記事で山根氏は「Macはおもちゃとしてなら十分遊べる。最近はビジネスにも使われるようになってきたが，ビジネスマシンとしては使いものにならない」といい切っています。Macファンの読者を相手に普通はMacファンの編集者が作る雑誌の記事として，このような明確な批判記事は異色といえましょう。

このような批判記事を載せることはきわめていいことだと思います。多様な意見を自由に発言し合うことこそが，もっとよいソフトウェア，もっとよいハードウェア，もっとよいサポート体制，もっとよい雑誌を実現する大きな力になるだろうと僕は思っているからです。ただし，署名つきであることと，反論も掲載することが条件です。

山根氏は，Macのどこが問題かという点で，特にワープロについて次のように批判しています。

「ワープロはどう考えてもワープロ専用機のほうが優れているからだ。Macintoshのように不安定な機械だけで原稿を書いている者はプロではないので，その意見をプロが参考にしては危ないものになる」

さらに，山根氏自身は書斎での原稿書きにはキヤノンのEZPSというマシンを使っているが，システムエラーが4年間1度もない，シャープのノートワープロWV-700は本体が1kgで，電池は長寿命で電源を入れた瞬間にシステムが立ち上がるなどと書いています。

そして，Macが進化して仕事に使われるようになるにつれて，Macのトラブルによって仕事を失い，重い赤字を抱えることになると警告しています。

強引な一般化

この記事の中身を吟味することにしましょう。まず，気がつくのは，Macのワープロは使いものにならないとしたうえで，いきなり不安定なMacはビジネスには使えないと強引な一般化を行っている点です。ワープロに関しては専用ワープロにかなわないと仮定しても，そのほかの分野のMacならではの，ほかに代替物のないようなさまざまな機能，それに関してはどうなのでしょうか。ビジネスの一部として実際に恩恵に与っている（いうまでもなく，Macのトラブルによって仕事を失ったり，重い赤字を抱えるようになったりしていない）数多くの人に対してもそう主張するのでしょうか？

でも，この点に関してはあまりつっこむのはやめることにしましょう。「ビジネスマシンではなかったMac」というタイトルをつけている山根氏自身がMacをビジネスマシンとして十分に活用してきたことは同じ連載の7月号を読めばわかります。

「デザイナーにとってはMacのグラフィック機能を失ったら仕事ができないだろうし，音楽家にとってはMacのサウンド機能なしには仕事にならないだろう。同じように，もの書きにとっては，Macのデータベースやアウトラインプロセッサを知ってしまったら，これなしに仕事ができなくなる」と書いているのです。

そうです。山根氏は，仕事に必要不可欠なマシンであることは十分に承知のうえで，あえてこのような一般化を行っているのです。ですから，この点に関しては，反論してもあまり実のある議論にならないような気がします。ビジネスマシンであるMacが

プロはMacを使わない?

なければ，山根氏の著作『ギャルの構造』は生まれなかったとまで同じ7月号で率直に書いているのです。

山根氏の不幸な事故

むしろ，この記事の中心である，Macのワープロなどは専用ワープロと比べるまでもないという主張に，注目すべきでしょう。

Macのワープロが優れていない点については，Macが安定していない（システムダウンしやすい）ことを最大の理由として挙げています。それだけなのでしょうか？

今回の前の4カ月分ほどを費やして，山根氏はある事件を何度も何度もとり上げています。それは，山根氏が愛用していたPowerBook100のハードディスクがクラッシュして駄目になったという同情すべき事件です。

山根氏はハードディスクの中には貴重なデータが入っているという理由でディスクごとメーカーに修理に出すことは拒否し，友人にハードディスクを抜き出してもらってから修理に出したところ，中身を開けたMacの修理は受けつけられなかったというのです。そしてその結果，なんと驚いたことにそのPowerBook100は捨てたとのこと（ウーム，これがプロ根性なのか？）。

また，ワープロに関しては，7月号を見ると，「Macで書くと2倍はかかる。これは仕事上，死活問題である」としており，安定云々の話は出ていません。

一方，昔の山根氏の本（文献2）がたまたま手元にありまして，その中で，「仕事で使いたいというのであれば，日本語ワープロ（専用機）を買ったほうがよい」と書いています。その理由については，「キーボードの配列などの点で日本語ワープロ専用機にはかなわないと肝に命じないと後悔する」としています。

これらの経緯を見ると，山根氏はパソコンのワープロ機能については，信頼性以外の操作性やそのほかの点に関してもずいぶんと昔から，劣るものだという意見をもっており，それに今回のPowerBook100の故障で，ついにその思いが決定的になったのであるということが推測されます。

山根氏所有のワープロ専門機を触る

山根氏は，ワープロ専用機よりMacのほうが劣ると断定しており，Macのワープロソフト同士の比較などまったく参考にもしない態度であることが読み取れます。しかし，Macのワープロソフトの処理速度や操作性などの問題は，ワープロソフトの出来や個性に左右されるところが少なくないので，専用ワープロかパソコンのワープロソフトか，という問題を考えるときには，個別のソフトを考慮した検討が必要なのはいうまでもないでしょう。

僕自身はMacのワープロはそこそこ知っているのですが，山根氏が太鼓判を押すワープロ専用機であるキヤノンのEZPS3500というのは見たことも聞いたこともありません。そこで，近所のキヤノン販売のショールームに押しかけていって，見せてもらうことにしました。インストラクトレス（こういう呼び方があったのかー！）の草野さん，ありがとうございました。

まずわかったことは，そのEZPS3500というのは何年も前の機種であり，ショールームには置いていないということです。かわりに，後継機種であるEZPS50とEZPS10が置いてありました。コピー機との接続などで制限はありますが，基本的な性能に関してはパワーアップされています。

次にわかったことは，これは重要なことなのですが，モデルによって違いますが，定価が200万円前後から350万円ぐらいになるということです。つまりレーザープリンタを入れたMacの値段に比べて5倍から10倍近くもするということです。

さすがに，DTP機ということだけあって，レイアウト，網掛け，書体の変形などの方面は目をみはるものがありました。どの機能も，探せばMacの各種のソフトとして個別に実現されているともいえますが，それらを集大成した感じです。また，プリンタの印字もさすがキヤノンです。

では，そのようなDTP的な機能ではな

く，単に高速に文章を入力し，編集したいという場合には，はたしてどうなのでしょう？　これは微妙な問題であると思えます。たとえば，この専用機の場合，ページの概念がまずあるので，画面をスクロールすることはできず，2ページ目3ページ目と切り離して表示されるのです。

1ページ目のしっぽのほうを見ながら3ページ目を書きたいこともあるでしょうし，ざっと文章の頭からひととおり流し読みしたいこともあるのに，スクロールすることができないのです。いろいろレイアウトすると，かなり画面表示が重くなるので，スクロールするにはまだCPUパワーが不足するためなのでしょうか？

山根氏がなぜ，このマシンのほうが2倍早く文章が書けるといったのかは，1時間ちょっとの体験では理由がわかりませんでした。特別なキーボードで早い入力法を採用しているのかどうかはわかりませんが，そもそも，僕が説明を受けた機種より，山根氏のマシンはCPUが何割か遅いのです。

最後に，EZPSシリーズのユーザーについてたずねたところ，印刷関係の仕事の方(印刷するための版下を手軽に作る)が多く，次に，会社で社内報を作る部署などでも導入されているという答えがかえってきました。やはり，DTP機ということで，そういう出来上がり自体も重視されるところで強みを発揮しているのでしょう。

値段といい客層といい，Macか専用機かという比較はあまりに荒っぽい選択肢であるという印象を強めました。

オープンかクローズドか？

基本的な僕の考えは，山根氏のように(本当の主張や真意であるかどうかは定かではありませんが) Macはビジネスマシンではないと排除するのではなく，うまく使ったほうが必ず得ですよということです。同情すべきシステムダウンあるいはハードディスククラッシュについても，僕は問題はそう難しくないと考えます。

Macintoshのような不安定な機械だけで原稿を書くのはプロではないという表現がありましたが，確かにそういいたくなる気持ちも理解できます。しかし，計算機とうまくつき合っている人というのは，そういう痛い目にあいながらも，結局はごく簡単な防御策を覚えているのです。たとえば，4～5万でハードディスクを1つ余分に買ってきてつないで，こまめにバックアップをとるのはどうでしょう。バックアップったって，たとえば，ワープロによっては，何文字か打つごとに自動的に2つの場所にセーブしてくれますので，何も考えなくてもいいのです。

illustration：Haruhisa Yamada

専用か汎用かという問題に関していえば，専用マシンの場合は，スペシャライズして作られているわけですから専用がいいに決まっています（うーむ，理由になってないか？）が，経済原理が働くので，実際は必ずしもそうともいえません。専用機で何百万もするマシンのためにソフトを作るよりは，パソコン用にソフトを作ったほうが数が出て儲かるため，部分的にでも技術が進んでいるのは，むしろ安いワープロソフトである可能性だってあります。

ただし，汎用か専用かという場合に，山根氏が指摘する信頼性の問題は常につきまとってくるものであると思います。汎用にするということはシステムをオープンにすることです。オープンにしてさまざまなソフトが走るようにすれば，当然，予想もしないエラーが出るでしょうし，眠っていたバグを叩き起こすことにもなるでしょう。クローズドなシステムのほうが安全なのは傾向としては当然です。

Macの場合も，システムダウンの原因の多くは，アップルの予想もしなかったようなシステムパッチ (extension) によって引き起こされます。ですから，余分なソフトを一切入れなければ，信頼性は大幅に上げることができます。

ですから，システムダウンが気になる場合には，システムダウンの許されない削ぎ落とされたMacと，いろいろチューンアップして凝るためのMacと2台用意して，仕事の種類に応じて使い分ければいいのです。もちろん，ハードディスクも2台つけましょう。

「Macはビジネスマシンではない」といまさらいっても，ビジネスマシンとしてすでにMacを使っている数多くの人は，この記事を読んでも，「ふーん」といって一瞬共感を覚えたのち，再びMacの前に戻ってバリバリと仕事を開始することでしょう。

文献
1）　山根一眞，「Oh! Lovely "Sad Mac"：ビジネスマシンではなかったMac」，Macworld12月号，1993.
2）　山根一眞，「スーパー書斎の仕事術」，ビジネスアスキー，1986.

愛読者プレゼント

1 スーパーリアル麻雀PII&PIII

ビング ☎03(5496)2501

X68000用 5"2HD版

12,800円(税別) 3名

先月号で紹介した3人娘との対戦麻雀ゲーム。パッケージには，ジグソーパズルやCDなどの楽しいオマケも入っています。

2 ぬりぐすり

5名

江口響子さんが香港で仕入れてきたぬりぐすりを3個セットで。

3 中国茶

3名

これも響子さんのお土産の中国茶を1個ずつ3名に。3種類あるけど，選択は担当者の独断によります。

4 ザウルス出現!

ソフトバンク ☎03(5642)8101

5名

1,500円(税込)

シャープより発売された新携帯情報ツール，液晶ペンコム「ザウルス」の活用本。SOFTBANK MOOKです。「ザウルス」については1993年11月号の「ペンギン情報コーナー」でもご紹介ずみですね。

5 MIYA-NET 特製カレンダー

埼玉県の草の根ネットMIYA-NETより1994年のカレンダーをプレゼント。毎月かわいいワンちゃんに会えますよ。

MIYA-NET ☎048(650)1234(9600,14400bps)は，Z-MUSIC-SIGがある草の根ネットです。ゲストIDでもファイル転送が可能です。ただしファイル転送は日中の回線のすいている時間を利用してください。

6 ソフトバンク卓上カレンダー

20名

ソフトバンクからは今年も卓上カレンダーです。こちらはかわいいパソコンのイラストです。

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1994年1月18日の到着分までとします。当選者の発表は1994年3月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

11月号プレゼント当選者

1ネクタイピン (兵庫県)進戸健太郎 2コットン (愛知県)成瀬昭紀 (兵庫県)菅谷英明 (山口県)冨田昌胤 3湯飲み (埼玉県)鈴木道明 (岐阜県)後藤史生 (奈良県)西岡山誠人 (大阪府)岩室元典 (沖縄県)金城正之 4レーザーアクティブのすべて (千葉県)小笠原浩一 (東京都)池崎友博 (奈良県)田中謙一 (岡山県)野崎国彦 (福岡県)藤原正浩 (敬称略)
以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

戦略核兵器で遊ぶ

NUCLEAR WAR

Kojima Keizou 小島 慶三

ヨーロッパを舞台にした，戦略核兵器ウォーシミュレーション「NUCLEAR WAR」。怖いくらい手軽に使える核兵器を駆使して，全都市制覇を目指しましょう。操作にはジョイステックが必要です。

先日，街を歩いていると「核兵器追放運動」に出くわしました。確かに，核兵器がなければ安心して暮らせます。核兵器を知らない人はいないと思いますが，要するにあの有名な原子爆弾のことです。

まず，ウラン235などの分裂可能な物質の原子核に，中性子をたたき込み分裂させます。このときに，エネルギーと中性子が放出され，その中性子がほかの原子核に当たればさらに分裂を引き起こします。そのため，ある程度のウラン235を集めておくと連鎖反応が可能となり，高エネルギーが得られます。これを利用したのが原子炉であり原子爆弾なのです。

ここではその核兵器を使用したシミュレーションゲームを紹介します。このゲームに使用される核兵器は，2000km以上離れた都市に正確に命中させることができ，威力が少し弱い核爆弾です。これはゲームなので核反対派の人も気軽に遊んでください。

入力方法

このプログラムはX-BASICで書かれているので，X-BASICを起動して入力するか，またはエディタで入力してください。ダブルクォーテーション内のスペースも正確に入力するようにしてください。

なお，コンパイルはできません。インタプリタ上でも問題なく動く速度なので，まあ，大丈夫でしょう。もしも，カーソルの移動の速さが気になる人は，700行のcの数値で調節するといいでしょう。

遊び方&ゲーム進行

最大3人まで同時プレイが可能です。プレイヤーはイギリス軍，ドイツ軍，フランス軍の司令官となり，全都市を占領することが目的です。

操作は名前の入力以外はすべてジョイスティック1で行います。十字キーでカーソルの移動や数値の増減，Aボタンで決定，Bボタンでキャンセルまたは最下段のコマンドの実行です。ただし，委任コマンドの操作は少し違うので，これは次のコマンド解説を参照してください。

司令官は，毎月，各都市に命令を出してその都市の残り実行回数が0になるか，または，PASSをするとその都市の戦略は終了となります。

また，毎月兵隊に給料を支払います。その額は，軍事力に比例しているので，軍事力が高ければよいというものではありません。3，6，9，12月には住民からの税金などの収入，地雷維持費の支出があります。

コマンド解説

コマンドの右の数字分だけ実行回数が減ります。

●侵略

白線で結ばれた他国の都市に侵略します。そして戦闘となるわけですが，あまり大袈裟なものはなく両軍の軍事力が減っていくだけで，シェルターの数や攻守は軍事力の減り具合に影響します。侵略した場合，Bボタンで退却します。しかし，侵略された場合，逃げることはできません。Aボタンで戦闘速度が増すので，両軍とも軍事力が高いときは利用しましょう。

●核攻撃

他国の都市を核爆弾で攻撃します。都市間の距離は関係ありませんが，1弾で10回攻撃するよりも，10弾で1回攻撃したほうが効果的です。

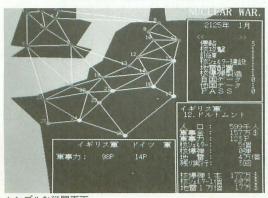
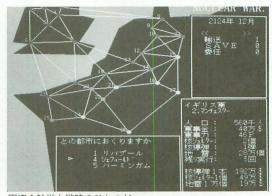
●訓練

使用した金額により，軍事力が上がります。

●核シェルター建設

核シェルターは侵略されたり，核攻撃されたときに有効です。都市の防御力のようなものです。

●地雷配置

地雷は侵略されたとき

シンプルな戦闘画面

軍資金輸送も戦略のひとつだ

に敵の軍事力を減らします。

●核爆弾製造

核爆弾を製造します。

●自国データ・他国データ

その国の全都市のデータを見ることができます。

●PASS

現在，戦略中の都市の戦略を終了します。残り実行回数は，制限つきで翌月に追加されます。

●輸送

白線で結ばれた自国の都市に，軍事力，軍資金を送ります。

●SAVE

現在の状態をディスクに保存します。続きはContinueでできます。

●委任

指定した都市を副司令官に任せます。このコマンドだけ操作方法が少し違います。

（都市の番号）＝（1－委任　0－解除）

となっていて，Aボタンで0・1の切り替え，Bボタンで終了です。すべての都市を委任して戦略を終了した場合，Bボタンを押し続ければ復帰できます。

戦略

このゲームの主役といったら変ですが，中心となるものは核爆弾です。これを使わなければ単なる都市盗りゲームになってしまいます。白線で結ばれた都市がすべて自軍の都市ならば，軍事力を上げても給料の額が増えるだけでほとんど意味はありません（危機に瀕している都市に輸送するのなら別）。輸送などで軍事力を減らし，核攻撃専門の都市にして，この都市から核攻撃をした直後に侵略すれば簡単に占領できるでしょう。

これは，基本的な戦略なので，それぞれ自分なりの戦略を立ててみてください。あと，ドイツ軍が少し強いようなので，この手のゲームは苦手という人は，ドイツ軍でプレイしてみるといいかもしれません。

最後に

このゲームの制作期間は約1週間，補修期間も約1週間で，急いで作ったため，非常に見づらいリストになってしまいました。変数もかなり複雑でグローバル変数とローカル変数が入り交じっているため，これ以上，手を加えないほうがいいでしょう。

では，ちょっとリストが長めですが，がんばって入力して楽しんでください。

イギリス各都市のデーター覧

愛しの核爆弾よ，飛んでけ～！

リスト1

```
 10 /*--------------------------------------------------
 20 /* NUCLEAR WAR. Programmed by K.Kojima
 30 /*                                  '93,9 '93,11
 40 /*--------------------------------------------------
 50 dim char mxy(63)={
 60      30,0,30,8,36,10,30,22,36,24,33,29,4,28,10,23,4,22,10,
 70      0,0,0,0,102,20,102,16,65,4,50,14,49,34,40,35,34,40,
 80      30,45,15,51,10,58,9,59,0,30,0,64,0,65,9,80,8,92,0,
 90      72,0,68,2,65,0,64,0 }
100 dim char txy(51)={ 27,12,42,11,
110     58,10,55,20,45,33,27,56,70,55,155,25,145,37,
120     150,48,175,48,135,70,126,82,144,92,170,75,150,110,
130     170,123,100,98,71,110,86,118,110,116,97,132,44,128,
140     78,156,48,160,122,176 }
150 dim int tdt(27,14)={
160      543, 500,60, 3,1,30,1,3, 2, 5, 6, 0, 0, 0,0,
170      475, 420,50, 2,0,28,1,3, 1, 4, 5, 0, 0, 0,0,
180      727, 510,56, 3,1,32,1,3, 4, 8, 0, 0, 0, 0,0,
190      553, 600,62, 3,1,33,1,3, 2, 3, 5, 7, 8, 0,0,
200     1122, 700,65, 5,2,50,1,3, 1, 2, 4, 6, 7, 0,0,
210      413, 340,52, 2,0,30,1,3, 1, 5, 7,19,23, 0,0,
220     6675,2200,60,30,2,50,1,3, 4, 5, 6,13,18,19,0,
230     1755, 720,65, 8,1,30,2,3, 3, 4, 9,11, 0, 0,0,
240      555, 450,55, 2,0,31,2,3, 8,10,11, 0, 0, 0,0,
250      532, 430,55, 2,1,32,2,3, 9,11,12,15, 0, 0,0,
260     3146,1720,66,15,2,50,2,3, 8, 9,10,12,15, 0,0,
270      605, 520,60, 3,0,35,2,3,10,11,13,14,15, 0,0,
280     1302, 650,62, 7,2,46,2,3, 7,12,14,18, 0, 0,0,
290      631, 450,55, 4,2,30,2,3,12,13,15,16,18,21,0,
300      579, 500,58, 3,1,31,2,3,10,11,12,14,16,17,0,
310      575, 470,59, 3,1,28,2,3,14,15,17,21, 0, 0,0,
320     1307,1010,61, 6,2,26,2,3,15,16,26, 0, 0, 0,0,
330      675, 420,49, 2,1,32,3,3, 7,13,14,19,20,21,0,
340      627, 410,52, 3,2,41,3,3, 6, 7,18,20,23, 0,0,
350      613, 450,55, 3,0,35,3,3,18,19,21,22,23,24,0,
360      765, 430,55, 4,1,31,3,3,14,16,18,20,22, 0,0,
370     2834,2060,66,10,3,51,3,3,20,21,24,26, 0, 0,0,
380      523, 420,62, 2,3,63,3,3, 6,19,20,24,25, 0,0,
390      506, 400,52, 2,0,40,3,3,20,22,23,24,25,26,0,
400      512, 350,51, 3,1,28,3,3,23,24, 0, 0, 0, 0,0,
410      712, 450,51, 4,2,43,3,3,17,22,24, 0, 0, 0,0,
420      160,40,10,0,0,0,0,0,0,0,2124,4}
430 dim str nam(2)={"イギリス","ドイツ  ","フランス"}
440 dim str cm(45)[22]={"はい ","いいえ","1人","2人"
450     ,"3人","Continue","Start"
460     ,"人 口 :          千人","軍資金 :          万S"
470     ,"軍事力 :          P  ","核シェルター:          個  "
480     ,"核爆弾 :          弾  ","地 雷 :          万個"
490     ,"残り実行:          回  ","侵略          3"
500     ,"核攻撃          1","訓練          1","核シェルター建設・1"
510     ,"地雷配置     1","核爆弾製造  1","自国データ  0"
520     ,"他国データ  1","PASS      0","輸送          1"
530     ,"SAVE      0","委任          0","100%"
540     ,"   75%","   50%","   25%","<UP >DOWN (1-"
550     ,"ボーン","ドッカーン","成功しました","大成功です"
560     ,"大成功です!","大成功です!!!"
570     ,"はほとんどありません","は少ないようです"
580     ,"があります","は大きいようです","は大きいようです!"
590     ,"は非常に大きいようです","核爆弾1本          万S"
600       ,"核シェルター1個          万S","地雷1万個          万S" }
610 dim str nam3(26)[12]={"リバプール  ","マンチェスター        ",
620         "リーズ          ","シェフィールド        ","バーミンガム",
```


▶「ファイル共有の実験と実践」はOh!MZの香りがする。　倉田　泰幸(23)茨城県

```
630         "ブリストル ","ロンドン   ","ハンブルク ",
640         "ブレーメン ","ハノーバー ","ベルリン   ",
650         "ドルトムント","ケルン&ボン","ﾌﾗﾝｸﾌﾙﾄ     ",
660         "ライプチヒ ","ｼｭﾂｯﾄｶﾞﾙﾄ    ","ミュンヘン ",
670         "リール     ","ルアーブル ","ルアン     ",
680         "ランス     ","パリ       ","レンヌ     ",
690         "オルレアン ","ナント     ","リヨン     "}
700 int i,j,n,n1,n2,n3,n4,n5,n6,n7;n8,n9,na,s,st,a,b=1,c=230
710 dim int sortx(26),sorty(26),sv(500)
720 str nam2(3)[10],as[10],bs[210],cs[200]
730 screen 1,1,1,1:console 0,32,0:palet(1,0)
740 BGMSET():vpage(0):MAKEMAP():SETLINE():TITLE():vpage(15)
750 SETCITY():PR3():PRCOM(44,19,7,7):b=4
760 BGM():PRCOM(44,27,43,3):PR5()
770 repeat:SENRYAKU():until n4>400
780 if n4>700 then ENDING1() else ENDING2()
790 end
800 /*----------------------------------------------------
810 func MAKEMAP():apage(3) /*-------------------------------
820  fill(0,0,511,511,8)
830  for i=0 to 30:j=i*2
840   line(mxy(j)*5,mxy(j+1)*5,mxy(j+2)*5,mxy(j+3)*5,3)
850   if i=22 then i=i+1
860  next
870  paint(10,10,3):paint(350,11,3)
880  fill(344,260,511,511,0):fill(102,360,344,511,0)
890  box(346,263,509,509,15):box(104,362,343,509,15)
900  box(347,264,508,508,14):box(105,363,342,508,14)
910  fill(364,40,511,260,0)
920  box(366,42,509,260,15):box(367,43,508,259,14)
930  symbol(368,9,"NUCLEAR WAR.",1,1,2,11,0)
940 endfunc
950 func SETLINE():apage(3) /*-------------------------------
960  for i=0 to 25:for j=0 to 5
970   if tdt(i,j+8)<>0 then {n=tdt(i,j+8)*2-2
980  if n<52 and n>i then {
990  line(txy(i*2)*2,txy(i*2+1)*2,txy(n)*2,txy(n+1)*2,15)}}
1000  next:next
1010 endfunc
1020 func SETCITY():apage(3) /*-------------------------------
1030  for i=0 to 25
1040   for j=0 to 4
1050   circle(txy(i*2)*2,txy(i*2+1)*2,4,tdt(i,6)*2+3,,,j*64)
1060   next
1070   symbol(txy(i*2)*2-15,txy(i*2+1)*2-13,right$(" "+str$(i+1
),2),1,1,0,13,0)
1080  next
1090 endfunc
1100 func TITLE() /*--------------------------------------------
1110 apage(1):for i=0 to 1
1120  symbol(39+i,51-i,"NUCLEAR WAR.",3,4,2,11-i,0):next
1130  vpage(7):color 5:PRCOM(22,19,5,2):color 7
1140  locate 21,29:print "@1993.9.13 ":color 3
1150  repeat
1160  COP(158,322,1)
1170   switch n4
1180    case 0:n5=DLOAD():break
1190    case 1:OPENING():n5=0:break
1200   endswitch
1210  until n5<>-1
1220  apage(1):fill(0,0,511,511,0):cls
1230 endfunc
1240 func COP(n1,n2,n3) /*--------------------------------------
1250  YAJIRUSI(n1,n2):n4=0
1260  repeat:BGM()
1270   s=stick(1):st=strig(1)
1280   if s mod 6=2 then {n4=(n3+1+n4-(s=2)+(s=8)) mod (n3+1)
1290   home(0,0,511-n4*16):BP():WAIT3(570) }
1300  until st<>0:if st=2 then n4=n3
1310  fill(n1,n2,n1+8,n2+8,0):BP()
1320 endfunc
1330 func OPENING():repeat /*------------------------------------
1340  cls:locate 20,14:print "何人でプレイしますか?"
1350  PRCOM(22,15,2,3)
1360  COP(164,258,2):tdt(26,3)=n4+1:cls
1370  for i=0 to tdt(26,3)-1
1380   locate 20,14:print "プレイヤー"+str$(i+1)+"の名前は"
1390   locate 21,16:linput "10字まで>>";as
1400   nam2(i)=left$(as+space$(10),10)
1410   locate 25,16:print space$(12)
1420   next:if tdt(26,3)<3 then {nam2(tdt(26,3))="COM1   "
1430   if tdt(26,3)=1 then nam2(2)="COM2   "
1440  }
1450  cls:for i=0 to 1
1460  locate 20,14:print nam(i);"軍司令官は誰ですか?  "
1470 for j=0 to 2:locate 22,16+j:print nam2(j)
1480 next
1490 COP(165,258,2)
1500  if tdt(26,4+n4)<>0 then i=i-1 else { tdt(26,4+n4)=i+1
1510  tdt(26,7+i)=n4:locate 34,16+n4:print nam(i)+"軍司令官"}
1520  next
1530  for i=0 to 2
1540   if tdt(26,4+i)=0 then { tdt(26,4+i)=3:tdt(26,9)=i
1550   locate 34,16+i:print nam(2)+"軍司令官" }
1560  next:locate 21,20:print "これでよろしいですか?"
1570  PRCOM(22,21,0,2)
1580  COP(158,354,1)
1590  if n4=1 then for i=0 to 2:tdt(26,4+i)=0:next else i=14
1600 until i=14
1610  cls
1620  locate 8,14:print "2124年  この頃、「平和」なんて言葉は存在しない。"
1630 print "    毎日、どこかで戦争が起きていた。この年、三大強国の"
1640 print "       イギリス、ドイツ、フランスによる、 三国核戦争が勃発"
1650 print "   した ・・・"
1660 WAIT2():SORTC()
1670 endfunc
1680 func WAIT() /*---------------------------------------------
1690  repeat
1700   st=strig(1+a+(a=2)):BGM()
1710  until st=0
1720 endfunc
1730 func WAIT2():WAIT() /*-------------------------------------
1740  repeat
1750   st=strig(1+a+(a=2)):BGM()
1760  until st<>0 or a<>0
1770 endfunc
1780 func PRCOM(n1,n2,n3,n4) /*---------------------------------
1790  int i:for i=1 to n4
1800   locate n1,n2+i:print cm(n3+i-1)
1810  next:BGM()
1820 endfunc
1830 func SORTC() /*--------------------------------------------
1840  int i:for i=1 to 26
1850  n1=rnd()*25+1
1860  repeat
1870   if sortx(n1)<>0 then n1=(n1+1) mod 27
1880  until sortx(n1)=0
1890  sortx(n1)=i:next
1900 endfunc
1910 func SENRYAKU() /*-----------------------------------------
1920  st=strig(1):sn=0
1930  repeat:sn=sn+1:until sortx(sn)<>0:sm=sortx(sn)-1
1940  bs=right$(" "+str$(sm+1),2)+"."+nam3(sm)+"戦略"
1950  if tdt(26,6+tdt(sm,6))+tdt(sm,14)*3<tdt(26,3) or (st<>0 a
nd tdt(sm,14)=1) then {
1960  bs=bs+space$(37)+nam2(tdt(26,6+tdt(sm,6)))
1970  bs=bs+"司令官  命令を":tdt(sm,14)=0
1980  PR2():SENPLA() } else bs=bs+"中":PR2():SENCOM()
1990 sortx(sn)=0
2000 if sn=26 then MONTH():SORTC()
2010 endfunc
2020 func PR() /*-----------------------------------------------
2030  PR2():WAIT2()
2040 endfunc
2050 func PR2() /*----------------------------------------------
2060  if a=0 then {  CL(14,23,28,8)
2070  for n=1 to strlen(bs)/28+1:BGM()
2080   locate 14,22+n:print mid$(bs,(n-1)*28+1,28)
2090  next }
2100 endfunc
2110 func PR3() /*----------------------------------------------
2120  locate 50,3
2130  print using "####年 ##月";tdt(26,10),tdt(26,11)
2140 endfunc
2150 func PR4(n1) /*--------------------------------------------
2160 locate 44,17:print nam(tdt(n1-1,6)-1)+"軍"
2170 locate 45,18:print right$(" "+str$(n1),2)+"."+nam3(n1-1)
2180 for i=0 to 6
2190  locate 53,20+i:print using "######";tdt(n1-1,i-(i=6))
2200 next
2210 endfunc
2220 func SENCOM() /*-------------------------------------------
2230 a=1:na=0:for i=0 to 5:sorty(i)=0:next
2240 repeat:if tdt(sm,1)>tdt(26,2) then {
2250  if tdt(sm,2)<50+na*10 then {
2260  if tdt(sm,1)>25 then KUNREN()
2270  } else if tdt(sm,5)<50+na*5 then {
2280  if tdt(sm,1)>tdt(26,2)*2+9 and tdt(26,2)<15 then JIRAI()
2290  } else if tdt(sm,0)>tdt(sm,3)*(200-na) and tdt(sm,3)<10+n
a then {
2300  if tdt(sm,1)>tdt(26,1)+9 and tdt(26,1)<50 then SHELKEN()
2310  }
2320  if tdt(sm,7)>0 and na>20 then {SORTY()
2330  if n1=0 then {
2340  if tdt(27,tdt(sm,6)-1)<>0 and tdt(sm,4)>0 then {
2350  if tdt(tdt(27,tdt(sm,6)-1)-1,6)<>tdt(sm,6) then {
2360  n4=tdt(27,tdt(sm,6)-1):KAKUKOU()/*:tdt(27,tdt(sm,6)-1)=0
2370  }} else {
2380  for i=1 to n2:for j=0 to 5
2390  if tdt(sorty(i-1),8+j)>0 then {
2400  if tdt(tdt(sorty(i-1),8+j),6)<>tdt(sm,6) or i=n2 then {
2410  n4=i-1:n8=i:n9=j:YUSOU():j=n9:i=n8 }}
2420  next:next }} else {for i=0 to n1-1
2430  if tdt(sorty(i),2)>tdt(sm,2)*1.1#+50-na then {
2440  if tdt(sm,4)>0 and tdt(sm,7)>0 then {
2450  n4=sorty(i):n9=i:na=n1:KAKUKOU():n1=na:i=n9:na=90
2460  } else tdt(27,tdt(sm,6)-1)=sorty(i)
2470  }
2480  next }}
2490  if tdt(sm,7)>0 and tdt(sm,4)<na+5 and na<90 then {
2500  if tdt(sm,1)>tdt(26,0) and tdt(26,0)<190 then KAKUSEI()
2510  }}
2520  na=na+10
2530 until na>80 or tdt(sm,7)=0
2540 if tdt(sm,7)>3 then {SORTY()
2550 if n1>0 then {for i=1 to n1
2560 if tdt(sm,2)*(4-n1/2)/4>tdt(sorty(i-1),2)*1.1#+10 then {
2570 n4=i-1:MMP():SINRYAKU():i=6
2580 } else tdt(27,tdt(sm,6)-1)=sorty(i-1)
2590 next }}
2600 a=0:BGM():if tdt(sm,1)<0 then tdt(sm,1)=0
2610 endfunc
2620 func SENPLA() /*-------------------------------------------
2630 PR4(sm+1):PR5():locate 48,5:print "<<          >>"
2640 repeat:COP2()
2650 switch n4
2660  case 0:SINRYAKU():break
2670  case 1:KAKUKOU():break
2680  case 2:KUNREN():break
2690  case 3:SHELKEN():break
2700  case 4:JIRAI():break
2710  case 5:KAKUSEI():break
2720  case 6:PRDATA(tdt(sm,6)-1):break
2730  case 7:TADATA():break
```

▶わーい、X68000で「ぷよぷよ」が出るよー。でもその前に増設RAMだな……。
松本　亮(17)広島県

```
2740  case 8:n4=23:break
2750  case 9:YUSOU():break
2760  case 10:DSAVE():break
2770  case 11:ININN():break
2780 endswitch
2790 if n4<>6 and n4<>8 and n4<400 then CL(14,23,28,7)
2800  PR4(sm+1):until n4=23 or n4>400
2810 endfunc
2820 func PRDATA(n1):apage(2) /*-----------------------------
2830 fill(3,25,340,308,1):box(5,27,338,306,15)
2840 box(6,28,337,305,14):locate 4,2:print nam(n1)
2850 print "   都市名       人口  資金  軍 ｼｪﾙ 核  地"
2860 n2=0:for i=0 to 25
2870 if tdt(i,6)=n1+1 then {locate 1,5+n2
2880 print using "&              & ##### ##### ### ### ## ### ";nam
3(i),tdt(i,0),tdt(i,1),tdt(i,2),tdt(i,3),tdt(i,4),tdt(i,5);
2890 if tdt(i,14)=1 then print "ｲ"
2900 n2=n2+1 }
2910 if n2=13 and i<25 then {
2920  n2=0:color 7:print:print "   PUSH A"
2930  WAIT2():color 3:CL(1,5,42,14) }
2940 next:color 7:print:print "   PUSH A":color 3
2950 WAIT2():CL(1,2,42,17):fill(3,25,340,308,0)
2960 endfunc
2970 func CL(n5,n6,n7,n8) /*-----------------------------
2980 for n=1 to n8
2990  locate n5,n6-1+n:print space$(n7)
3000 next
3010 endfunc
3020 func MONTH() /*---------------------------------
3030 tdt(26,11)=tdt(26,11) mod 12+1
3040 if tdt(26,11)=1 then tdt(26,10)=tdt(26,10)+1
3050 if tdt(26,11) mod 3=0 then {
3060  for i=0 to 25
3070   tdt(i,1)=tdt(i,1)+tdt(i,0)/3-tdt(i,5)/5+10
3080  next }
3090 for i=0 to 25
3100  tdt(i,1)=tdt(i,1)-tdt(i,2)/10
3110  if tdt(i,1)<0 then tdt(i,1)=0
3120  tdt(i,0)=tdt(i,0)+tdt(i,0)*0.02#+1
3130  tdt(i,7)=tdt(i,7)+(tdt(i,7)>10)*3+3
3140 next
3150 for i=0 to 2
3160 tdt(26,i)=10*pow((2-i)*2-(i=2),2)+4*(9-i*3-rnd()*(3-i)*6)
3170  if tdt(26,i)<1 then tdt(26,i)=1
3180 next:PR5():PR3()
3190 endfunc
3200 func KUNREN() /*---------------------------------
3210 int n:n=100+(tdt(sm,1)<99)*(99-tdt(sm,1))
3220 if a=0 then {
3230 bs="軍事力UPにいくら使いますか <UP >DOWN (0-"
3240 bs=bs+str$(n-1)+")":PR2():n1=COP3(0,n,0) } else st=1:n1=n
3250 if st=1 then {tdt(sm,1)=tdt(sm,1)-n1
3260 n1=(n1*1.2#-(tdt(sm,2)*n1*1.2#/999))/5
3270 if n1+tdt(sm,2)>999 then n1=999-tdt(sm,2)
3280 if a=0 then bs="軍事力が"+str$(n1)+"UPした。":PR()
3290 tdt(sm,2)=tdt(sm,2)+n1:MMP() }
3300 endfunc
3310 func MMP() /*---------------------------------
3320  tdt(sm,7)=tdt(sm,7)-1:if tdt(sm,7)=0 then n4=23
3330 endfunc
3340 func PR5() /*---------------------------------
3350 for i=0 to 2
3360  locate 55,28+i:print using "####";tdt(26,i)
3370 next
3380 endfunc
3390 func KAKUKOU() /*---------------------------------
3400  if tdt(sm,4)>0 then {
3410   bs="どの都市に撃ちますか"+space$(8)+cm(30)+"26)":PR2()
3420  if a=0 then n3=COP3(1,27,4) else n3=n4+1:st=1
3430  if st=1 then {
3440  if tdt(n3-1,6)<>tdt(sm,6) then {
3450  n2=10+(tdt(sm,4)<10)*(10-tdt(sm,4))
3460   bs="何弾撃ちますか?"+space$(13)+cm(30)+str$(n2)+")"
3470   PR2()
3480   if a=0 then n1=COP3(1,n2+1,1) else st=1:n1=n2
3490   if st=1 then{n5=txy(n3*2-2):n6=txy(sm*2):n7=txy(n3*2-1)
3500    n8=txy(sm*2+1):n2=sqr(pow(n5-n6,2)+pow(n7-n8,2))
3510    for i=1 to n1
3520     apage(0):home(0,0,0):circle(n6*2,n8*2,3,13)
3530     paint(n6*2,n8*2,13):circle(n6*2,n8*2,3,1)
3540     for j=1 to n2/2
3550      k=512-pow((j*4-n2),2)/(99+n2)+n2*n2/(99+n2)
3560      k=k-(n7-n8)*j*4/n2:if k>(n8*2)+509 then k=n8*2-3
3570      home(0,(512-(n5-n6)*j*4/n2) mod 512,k mod 512)
3580     BGM():WAIT3(20)
3590     next:fill(n6*2-5,n8*2-5,n6*2+5,n8*2+5,0)
3600    circle(n6*2,n8*2,20,5,0,180,356)
3610    circle(n6*2,n8*2,15,5,180,0,160)
3620    paint(n6*2,n8*2,5):m_play(2):WAIT3(1250)
3630    fill(0,0,511,511,0):next
3640  n2=tdt(n3-1,3):bs="":n5=0:if a=1 then {
3650   cs=nam(tdt(sm,6)-1)+"軍が"+right$("  "+str$(n3),3)+"."
3660   cs=cs+nam3(n3-1)+"に核爆弾を撃ちました。"+space$(34)
3670  if tdt(26,6+tdt(n3-1,6))<tdt(26,3) then { a=2
3680  cs=cs+nam2(tdt(26,6+tdt(n3-1,6)))+"司令官"+space$(40)+"被害"
3690  }}
3700  if rnd()*(200+n1*30+n2*2)>200-n1*20 and n2>0 then {
3710   bs="核ｼｪﾙﾀｰ を破壊しました          ":n2=n2-1
3720   tdt(n3-1,3)=n2:n5=1 }
3730  n4=tdt(n3-1,2)
3740  if rnd()*(500+n4/2+n1*30)>450-n1*15 then {
3750   bs=bs+"軍事力に大ダメージ!            ":n4=n4/(2.2#+rnd())
3760   n5=n5+1 } else n4=n4/(1+rnd())
3770  tdt(n3-1,2)=n4:n4=70+rnd()*25-n1*6
3780  if n4<65 then n5=n5+1
3790  if n4<40 then n5=n5+1
3800  tdt(n3-1,0)=tdt(n3-1,0)+(tdt(n3-1,0)>tdt(n3-1,3)*100)*(td
t(n3-1,0)-tdt(n3-1,3)*100)*(100-n4)/100
3810  tdt(n3-1,5)=tdt(n3-1,5)*n4/100:n4=tdt(n3-1,4)
3820  if rnd()*(379+n1*70+n4*6)>380-n1*20 and n4>0 then {
3830   bs=bs+"核爆弾を破壊しました           ":tdt(n3-1,4)=n4-1
3840   n5=n5+1 }
3850  if a=0 then bs=bs+cm(31+n5) else {
3860  if a=2 then bs=cs+cm(37+n5) else bs=cs
3870  }
3880  tdt(sm,4)=tdt(sm,4)-n1:MMP()
3890  if a mod 2=0 then a=0:PR():BGM()
3900  if a=1 then a=0:PR2():a=1:BGM():WAIT3(800)
3910  }} else {bs="自国です!":PR()
3920  }}} else bs="核爆弾がありません!":PR()
3930 endfunc
3940 func KAKUSEI() /*---------------------------------
3950 if tdt(sm,1)>=tdt(26,0) and tdt(sm,4)<99 then {
3960 if a=0 then {
3970  bs="核爆弾製造は1回1弾です。 ("+str$(tdt(26,0))+"万$)よろしいですか
?"
3980  PR2():PRCOM(24,27,0,2):COP(176,448,1) }
3990 if n4=0 or a=1 then {tdt(sm,1)=tdt(sm,1)-tdt(26,0)
4000  tdt(sm,4)=tdt(sm,4)+1:MMP()
4010  if a=0 then bs="核爆弾を製造しました。":PR()
4020 }} else {if tdt(sm,4)=99 then bs="核爆弾は99弾までです。" else b
s="軍資金が足りません!"
4030 PR() }
4040 endfunc
4050 func JIRAI() /*---------------------------------
4060 if tdt(sm,1)>=tdt(26,2) and tdt(sm,5)<999 then{
4070  n1=tdt(sm,1)/tdt(26,2):n1=n1+(n1>50)*(n1-50)
4080  n1=n1+(n1+tdt(sm,5)>999)*(n1-999+tdt(sm,5))
4090  if a=0 then { n2=1
4100  bs="地雷を何万個配置しますか     "+cm(30)+str$(n1)+")"
4110  PR2():n2=COP3(1,n1+1,5) } else n2=n1:st=1
4120  if st=1 then { MMP()
4130  tdt(sm,1)=tdt(sm,1)-tdt(26,2)*n2:tdt(sm,5)=tdt(sm,5)+n2
4140  if a=0 then bs=str$(n2)+"万個の地雷を配置しました。":PR()
4150 }} else {
4160  if tdt(sm,5)=999 then { bs="地雷は999万個までです。"
4170  } else bs="軍資金が足りません!"
4180  if a=0 then PR()
4190 }
4200 endfunc
4210 func SHELKEN() /*---------------------------------
4220 if tdt(sm,1)>=tdt(26,1) and tdt(sm,3)<999 then{
4230  n1=tdt(sm,1)/tdt(26,1):n1=n1+(n1>10)*(n1-10)
4240  n1=n1+(n1+tdt(sm,3)>999)*(n1-999+tdt(sm,3))
4250  if a=0 then { n2=1
4260   bs="核ｼｪﾙﾀｰを何個建設しますか    "+cm(30)+str$(n1)+")"
4270   PR2():n2=COP3(1,n1+1,6) } else n2=n1:st=1
4280  if st=1 then { bs=str$(n2)+"個の核ｼｪﾙﾀｰを建設しました。"
4290   tdt(sm,1)=tdt(sm,1)-tdt(26,1)*n2:tdt(sm,3)=tdt(sm,3)+n2
4300   MMP():PR() }} else {
4310   if tdt(sm,3)=999 then {bs="核ｼｪﾙﾀｰは999個までです。"
4320   } else bs="軍資金が足りません!"
4330  PR() }
4340 endfunc
4350 func COP2():n9=0:na=0 /*-----------------------------
4360 YAJIRUSI(378,98):PRCOM(49,5,14,9)
4370  repeat
4380   s=stick(1):st=strig(1):BGM():n1=9-6*n9
4390   if s mod 6=2 then {
4400   na=(n1+na-(s=2)+(s=8)) mod n1:BP():WAIT3(350)
4410   } else if s=6 or s=4 then { n9=(n9+1) mod 2
4420    na=na+(na>8-6*n9)*(na-2):WAIT3(63):CL(49,6,13,9)
4430    PRCOM(49,5,14+n9*9,9-6*n9)}
4440   home(0,0,511-na*16)
4450  until st=1
4460 fill(378,98,386,106,0):BP():n4=na+n9*9
4470 endfunc
4480 func TADATA() /*---------------------------------
4490 n1=0
4500  for i=0 to 2
4510   if tdt(26,4+i)<>tdt(sm,6) then sorty(n1)=tdt(26,4+i):n1:
n1+1
4520  next
4530  bs="どの国のデータを見ますか":PR2()
4540  for i=0 to 1
4550   locate 22,26+i:print nam(sorty(i)-1)
4560  next
4570  COP(152,416,1):n2=99
4580 if st=1 then PRDATA(sorty(n4)-1):MMP()
4590 endfunc
4600 func YUSOU() /*---------------------------------
4610 n5=0:for i=0 to 5:if tdt(sm,8+i)>0 then {
4620  if tdt(tdt(sm,8+i)-1,6)=tdt(sm,6) then sorty(n5)=tdt(sm,8
+i)-1:n5=n5+1
4630 }
4640 next
4650 if n5=0 then {bs="隣都市に自国はありません。":PR()
4660 } else { if a=0 then {bs="どの都市におくりますか":PR2()
4670  for i=1 to n5:locate 21,24+i
4680  print using "## &              &";sorty(i-1)+1,nam3(sorty(i-1
))
4690 next:COP(128,400,n5-1)
4700 } else st=1
4710 if st=1 then {for n7=0 to 1
4720  if n7=0 then bs="軍資金をいくら送りますか      "
4730  if n7=1 then bs="軍事力をどれだけ送りますか   "
4740 n2=tdt(sm,n7+1)+tdt(sorty(n4),n7+1)+1
4750 if n2>999 and n7=1 then n2=999-tdt(sorty(n4),2) else n2=td
t(sm,n7+1)+1
4760 bs=bs+"<UP >DOWN (0-"+str$(n2-1)+")":PR2():n1=0
4770 if a=0 then n1=COP3(0,n2,2+n7) else st=1:n1=(n2-1)/3
4780 if n7=0 then n6=n1
4790 if st=2 then n7=2
```

▶表紙の顔が一瞬ベガの顔に見えた。　　吉本　康孝(21)福岡県

```
 4800 next
 4810 if st=1 then {bs="輸送しました。":PR():n5=sorty(n4)
 4820 tdt(sm,1)=tdt(sm,1)-n6:tdt(n5,1)=tdt(n5,1)+n6
 4830 tdt(sm,2)=tdt(sm,2)-n1:tdt(n5,2)=tdt(n5,2)+n1:MMP() }}}
 4840 endfunc
 4850 func BGMSET() /*-----------------------------------
 4860  m_init()
 4870  for i=2 to 4:m_alloc(i,500):m_assign(i,i):next
 4880  m_trk(2,"o0@68v14 a4")
 4890  m_trk(3,"l32o7@29v15 c")
 4900  m_trk(4,"t120l16o3@68v14 |:3r4a4.aaa.r32a8:| r4a8aaaaaaar
8r16r64")
 4910 endfunc
 4920 func BGM() /*--------------------------------------
 4930 if m_stat(b)=0 then m_play(b)
 4940 endfunc
 4950 func BP() /*---------------------------------------
 4960  m_play(3)
 4970 endfunc
 4980 func SINRYAKU() /*---------------------------------
 4990 if tdt(sm,7)>2 then {SORTY()
 5000 if n1=0 then {bs="隣都市はすべて自国です。":PR()
 5010 } else {n5=n1:bs="どの都市に攻めますか":PR2()
 5020  if a=0 then {for i=1 to n5:locate 22,24+i
 5030  print using "##.&           &";sorty(i-1)+1,nam3(sorty(i-1
))
 5040  next }
 5050  if a=0 then COP(160,400,n5-1) else st=1
 5060  if st=1 then {bs="何%出動しますか":PR2():n6=n4
 5070  if a=0 then { PRCOM(22,24,26,4):COP(160,400,3)
 5080  } else n4=n5/2:st=1
 5090  if st=1 then {
 5100  n7=tdt(sm,2)*(4-n4)/4:n1=(tdt(sorty(n6),5)+n7)/10
 5110  if n1>tdt(sorty(n6),5) then n1=tdt(sorty(n6),5)
 5120  tdt(sorty(n6),5)=tdt(sorty(n6),5)-n1
 5130  tdt(sm,2)=tdt(sm,2)-n7
 5140  if n7<n1 then n1=n7-1
 5150  if a=1 then {
 5160  bs=nam(tdt(sm,6)-1)+"軍が"+right$("  "+str$(sorty(n6)+1),3
)+"."+nam3(sorty(n6))+"に攻め入みました！"
 5170  a=0:PR2():a=1
 5180  if tdt(26,6+tdt(sorty(n6),6))<tdt(26,3) then a=2
 5190  }
 5200  n8=txy(sorty(n6)*2):n9=txy(sorty(n6)*2+1)
 5210  n2=sqr(pow(n8-txy(sm*2),2)+pow(n9-txy(sm*2+1),2))
 5220  apage(0):home(0,0,0)
 5230  circle(txy(sm*2)*2,txy(sm*2+1)*2,3,13)
 5240  paint(txy(sm*2)*2,txy(sm*2+1)*2,13)
 5250  circle(txy(sm*2)*2,txy(sm*2+1)*2,3,1)
 5260  for i=0 to 4:for j=1 to n2
 5270  home(0,(512-(n8-txy(sm*2))*j*2/n2) mod 512,(512-(n9-txy(s
m*2+1))*j*2/n2) mod 512)
 5280  BGM():next:next:fill(20,0,400,430,0)
 5290  bs="地雷で"+nam(tdt(sm,6)-1)+"軍軍事力 "
 5300  bs=bs+str$(n1)+"P    DOWM!":n7=n7-n1:PR()
 5310  if a=2 then a=0:PR():a=2
 5320  bs="      "+nam(tdt(sm,6)-1)+"軍   "
 5330  bs=bs+nam(tdt(sorty(n6),6)-1)+"軍"+space$(28)+"軍事力:"
 5340  PR2():n1=0:if a=2 then a=0:PR2():a=2
 5350  repeat
 5360  st=strig(1-(a=1)):st=st+(st=2)*(a=2)*2
 5370  n7=n7-int(rnd()*1.7#)
 5380  tdt(sorty(n6),2)=tdt(sorty(n6),2)-int(rnd()*1.51#)
 5390  for i=0 to -80*(a<>1)+(st=1)*80:BGM():next
 5400  n7=-1*n7*(sgn(n7)=1):tdt(sorty(n6),2)=-1*tdt(sorty(n6),2)
*(sgn(tdt(sorty(n6),2))=1)
 5410  if a<>1 then { locate 24,25
 5420  print using "###P    ###P";n7,tdt(sorty(n6),2) }
 5430  if n7<tdt(sorty(n6),2)*1.1# and a>0 then st=2
 5440  if n7=0 or tdt(sorty(n6),2)=0 then st=2
 5450  until st=2
 5460  if n7=0 then bs=nam(tdt(sm,6)-1)+"軍は力つきた・・":st=0
 5470  if tdt(sorty(n6),2)=0 then {st=0
 5480  bs=nam3(sorty(n6))+"を占領しました。"
 5490  if a>0 then bs=nam(tdt(sm,6)-1)+"軍が"+bs
 5500  tdt(sorty(n6),7)=3:tdt(sorty(n6),5)=tdt(sorty(n6),5)/2
 5510  tdt(sorty(n6),2)=n7:tdt(sorty(n6),6)=tdt(sm,6)
 5520  tdt(sorty(n6),14)=0
 5530  tdt(sorty(n6),1)=tdt(sorty(n6),1)+500 }
 5540  if st=2 then if a=0 then {
 5550  bs="退却します・・":tdt(sm,2)=tdt(sm,2)+n7
 5560  } else bs=nam(tdt(sm,6)-1)+"軍が退脚しました・・"
 5570  if a>0 then a=0:PR2():a=1 else PR2()
 5580  SETCITY():MMP():MMP():MMP():if a=0 then WAIT2()
 5590 }}}}
 5600 CLR()
 5610 endfunc
 5620 func CLR() /*--------------------------------------
 5630 for i=0 to 2:tdt(27,3+i)=0:next
 5640 for i=0 to 25
 5650  tdt(27,2+tdt(i,6))=tdt(27,2+tdt(i,6))+1
 5660 next
 5670 for i=0 to 2
 5680  if tdt(27,3+i)=0 and tdt(27,6+i)=0 then {
 5690  bs=nam(i)+"軍が全滅しました。":n1=a:a=0:PR2():a=n1
 5700  BGM():WAIT3(3500):tdt(27,6+i)=1 }
 5710  if tdt(27,3+i)=26 then {
 5720   if tdt(26,7+i)<tdt(26,3) then n4=777+i else n4=444+i
 5730  }
 5740  if tdt(27,3+i)=0 and tdt(26,7+i)<tdt(26,3) then { n1=0
 5750  for j=1 to tdt(26,3)
 5760   n1=n1+tdt(27,5+j)
 5770  next
 5780  if n1=tdt(26,3) then n4=454
 5790  }
 5800 next
 5810 endfunc
 5820 func ENDING1() /*----------------------------------
 5830  bs=nam2(n4-777)+"司令官！！"+space$(8)+nam(n4-777)
 5840  bs=bs+"軍が全都市を占領しました！！"+space$(20)
 5850  bs=bs+nam(n4-777)+"軍の勝利です！！"
 5860  PR()
 5870  bs=str$(tdt(26,10))+"年"+right$(" "+str$(tdt(26,11)),2)
 5880  bs=bs+"月：三国戦争は"+nam(n4-777)+"軍の勝利に終わった。"
 5890  PR()
 5900  bs="THANK YOU FOR PLAYING!!"
 5910  bs=bs+space$(38)+"Programmed by K.Kojima"+space$(53)
 5920  bs=bs+"END":PR2()
 5930 endfunc
 5940 func ENDING2():bs="" /*----------------------------
 5950  if n4<>454 then {bs=nam(n4-444)
 5960  bs=bs+"軍が全都市を占領しました・・"+space$(48) }
 5970  bs=bs+"     GAME    OVER":a=0:PR()
 5980 endfunc
 5990 func ININN() /*------------------------------------
 6000  n1=0:for i=0 to 25
 6010  if tdt(i,6)=tdt(sm,6) then sorty(n1)=i:n1=n1+1
 6020  next
 6030  bs="どの都市を委任しますか":PR2()
 6040  for i=0 to n1-1:locate 16+(i/5)*5,25+(i mod 5)
 6050  print using "##=#";sorty(i)+1,tdt(sorty(i),14)
 6060  next:n2=0:n3=0:YAJIRUSI(120,400)
 6070 repeat
 6080 s=stick(1):st=strig(1)
 6090 if (s-1) mod 2=1 or st=1 then {
 6100  n2=5+(n3=n1/5)*(5*(n3+1)-n1)+n2-(s=2)+(s=8)
 6110  n2=n2 mod (5+(n3=n1/5)*(5*(n3+1)-n1))
 6120  n3=(n1/5+1+n3+(s=4)-(s=6)) mod (n1/5+1)
 6130  tdt(sorty(n2+n3*5),14)=(tdt(sorty(n2+n3*5),14)+st) mod 2
 6140  locate 19+n3*5,25+n2
 6150  print using "#";tdt(sorty(n2+n3*5),14)
 6160  home(0,511-n3*40,511-n2*16):BP():WAIT3(360) }
 6170  BGM():if st=1 then WAIT()
 6180  until st=2:WAIT():fill(120,400,128,408,0)
 6190 endfunc
 6200 func SORTY():n1=0:n2=0 /*-------------------------
 6210  for i=0 to 5
 6220  if tdt(sm,8+i)>0 then {n2=n2+1
 6230  if tdt(tdt(sm,8+i)-1,6)<>tdt(sm,6) then {
 6240  sorty(n1)=tdt(sm,8+i)-1:n1=n1+1
 6250  }}
 6260 next
 6270 endfunc
 6280 func DSAVE() /*------------------------------------
 6290  bs="よろしいですか?":PR2():PRCOM(22,24,0,2)
 6300  COP(160,400,1)
 6310  if st=1 and n4=0 then {for i=0 to 25:for j=0 to 8
 6320  sv(j+i*9)=tdt(i,j-(j=8)*6):next
 6330  sv(258+i)=sortx(i+1):next
 6340  for i=0 to 1:for j=0 to 11
 6350  sv(234+j+i*12)=tdt(26+i,j):next:next
 6360  n1=fopen("WARSAVE.DAT","c")
 6370  fwrite(sv,284,n1)
 6380  fwrites(nam2(0)+nam2(1)+nam2(2),n1):fclose(n1)
 6390  bs="SAVE終了しました。":PR() }
 6400 endfunc
 6410 func DLOAD() /*------------------------------------
 6420  error off
 6430  n1=fopen("WARSAVE.DAT","r")
 6440  error on:if n1<>-1 then {
 6450   fread(sv,284,n1)
 6460   freads(bs,n1):fclose(n1)
 6470   for i=0 to 25:for j=0 to 8
 6480   tdt(i,j-(j=8)*6)=sv(j+i*9):next
 6490   sortx(i+1)=sv(258+i):next
 6500   for i=0 to 1:for j=0 to 11
 6510   tdt(26+i,j)=sv(234+j+i*12):next:next
 6520   for i=0 to 2:nam2(i)=mid$(bs,i*10+1,10):next
 6530  } else locate 22,23:print "ファイルがありません！"
 6540  return(n1)
 6550 endfunc
 6560 func WAIT3(n1) /*----------------------------------
 6570  int i:for i=0 to n1*c/100:next
 6580 endfunc
 6590 func YAJIRUSI(n1,n2) /*----------------------------
 6600  apage(0):home(0,0,0)
 6610  line(n1,n2,n1+8,n2+4,15):line(n1,n2+8,n1+8,n2+4,15)
 6620  line(n1,n2,n1,n2+8,15):paint(n1+1,n2+2,5):WAIT()
 6630 endfunc
 6640 func COP3(n3,n4,n5):int n6 /*---------------------
 6650 dim str cp(6)[4]={"万$","弾","万$","P","","万個","個"}
 6660  locate 24,26:print cp(n5)
 6670  repeat:BGM()
 6680  s=stick(1):st=strig(1)
 6690  locate 20,26:print using "####";n6+n3
 6700  if n5>4 then { locate 29,26
 6710  print using "#####万$";(n6+1)*tdt(26,7-n5) }
 6720  if n5=4 then locate 24,26:print "."+nam3(n6)
 6730  if s=4 or s=6 then {
 6740  n6=(n4-n3+n6-(s=4)+(s=6)) mod (n4-n3):WAIT3(250) }
 6750  if (n5=2 or n5=3) and s mod 6=2 and n4-n3>99 then {
 6760  n6=((n6/100+(n4-n3)/100+1-(s=8)+(s=2)) mod ((n4-n3)/100+1
))*100+n6 mod 100
 6770  WAIT3(250) }
 6780  if n6>n4-1 then n6=n4-1
 6790 until st<>0 or a=1
 6800 return(n6+n3)
 6810 endfunc
```

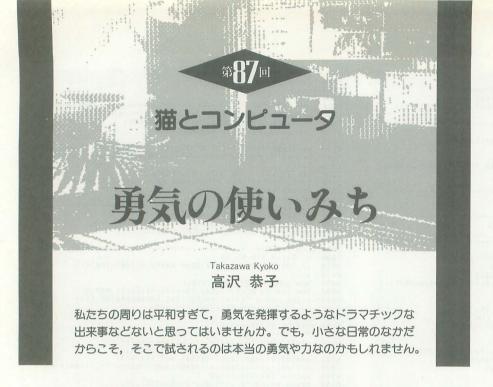
▶最近，毎朝必ず「からあげくんRED」を食べている。あのピリッとした味がいいんだよな。黄色よりやっぱりREDだな。　　　　菅谷　英明(27)兵庫県

第87回

猫とコンピュータ

勇気の使いみち

Takazawa Kyoko
高沢 恭子

私たちの周りは平和すぎて，勇気を発揮するようなドラマチックな出来事などないと思ってはいませんか。でも，小さな日常のなかだからこそ，そこで試されるのは本当の勇気や力なのかもしれません。

三毛猫の姉妹がいる酒屋さんの店先に，ちかごろ彼女たちの姿が見えなくなったので，いつも店番をしているおばあさんに聞いてみたいと思っていた。

スーパーマーケットに行く道すがら店をのぞくと，2匹はかならずそろって日だまりのなかにいた。姉妹と決めたのは，三毛猫のオスというのは，まずいないものと思うのが常識のようだからだ。

店は照明を節約しているのか昼も夜も薄暗く，なかも雑然としていて，いまどきの店にしてはあまりきれいではない。いつもいるのは，中年のご主人とその母親らしいおばあさんの2人，それと三毛猫の彼女たちだった。

おばあさんの暗算

ここは便利な町で，わが家から自転車で3分ほどのところだけでも大小のスーパーマーケットが5，6軒はあり，私もその日によって出かける店がちがうというぐあいだから，その酒屋さんの前をそうひんぱんに通るわけでもない。

しかしこのごろはいつのぞいても猫たちがいないので，なんとかたずねるチャンスをつくりたいと思っていたら，なんとこんどは，おばあさんも見あたらなくなってしまった。

タバコ売り場のある窓ぎわに悠然とすわっいた大柄のおばあさん。売り場に無頓着におかれたダンボール箱をおのおのが選び，それぞれのポーズでその上で昼寝をする一対の置物のような猫たち。ほの暗い店によくとけこんで，ちょっと時代おくれの感じがなかなかよかったものだ。

その時代おくれにひかれて，おばあさんがひとりでいるときをねらって(?)買い物をしてみたこともある。薄暗い店でおばあさんと話をしてみたかった。

小さなレジスターはあるが，おばあさんは使えないのか，あるいは防犯上，息子さんが使わせないのか，売り上げは暗算で勘定しているらしい。私は感心してしまって，「毎日暗算をしているお年寄りなんてめったにいるものじゃない，おばあさんはけっしてボケることなんかないでしょうね」というと，とても得意そうに，もう84歳だと自分から教えてくれた。

ところがそのあと，私が缶ビールを買うために3種類くらいの銘柄を陳列ケースから出してきて帳場に並べたら，どうも計算がむずかしくなったようだ。おばあさんをひとりにしておいていいのかなと思いながら，私がそばにあった紙きれとマジックインキで値段を聞きながら計算書をつくりはじめたとき，奥からけわしい顔でご主人があらわれた。どうやらおばあさんは，息子さんの監視つきで店番あそびをしているようでもあった。

みりんを買うことにして店に入り，「おばあちゃんは？」とあっさり聞いてみた。急に元気がなくなって，寝こんだままだと息子さんが教えてくれた。ついでにと思ったがとうとう猫のことは聞けなかった。2つともよくない知らせなら，息子さんも楽しくないだろう。

ただ，おばあさんも猫も展示できなくなったほの暗い店は，薄暗いだけの酒屋さんになりそうでさみしい。

地下鉄の車内にて

ホンニャアのほうはいたって元気といいたいところだけれど，やっぱり若猫のころとはちがう。争いも攻撃もひかえているようだ。このごろでは仰向けに眠らなくなったのも，いいトシをしてみっともないと考えたからではなく，なにか事態が急変したときとっさに対応できる自信がなくなったためではないだろうか。

若さの特徴のひとつは，強さをもっていることだ。やるべきと思ったことを行動できて，いためつけられても立ち直れるだけの力があることだろう。

医学部をめざしている友人の白井君と2人で，トオルが地下鉄に乗っていたときのこと。車内はすいていて，立っている人はトオルたちをふくめてわずかだった。すこし離れたところの座席で，となりあった同士がなにやら言い合いをしていたのだが，竹橋にさしかかったあたりでとうとうつかみ合いがはじまった。白井君は持っていた剣道の竹刀と胴着の入った袋をトオルに手渡すと，つかつかと争っている2人の席まで行き，「おやめなさい」と一喝して双方をひきはなした。

一見，立派な会社員ふうだった2人のおとなは，トオルがいうには，白井君の凛とした態度と，制服の高校生にたしなめられた恥ずかしさからか，すぐに争いをあきらめたという。

「カッコよかったなぁ」とトオルが帰宅していったとき，私はつい，「なにごともなかったからいいけれど……」とつぶやいてしまったが，彼にはそれが気にいらなかったようだ。すこしムッした顔で，「ボクもやるべきときにはやると思うから，お母さんたちもそんなことで後悔しないでほしいんだ」といった。

説明はなかったが，トオルがカッコイイと感じたのは，白井君がおとなたちの争いをやめさせたことではなくて，正しいと思ったことをまよわずに行動したことだ。たまたま白井君は剣道二段の資格をもってい

たが，それが技としてではなく意志の強さとしてあらわれたと感じられたことも，またカッコよかったのだろう。

やらねばならない，見逃しておけないというときにいちばん必要なのは，まよわないことではないか。ひきかえに失うものが気になる人はダメだ。不愉快な思いをしたくない。危害をこうむりたくない。命を失いたくない。まよわない人には強さがあるからそれが相手にも伝わり，よい結果になることもあるだろうが，ときには逆に相手をなお刺激してしまい，悪い結果を生むこともあるのだろう。

M君の追究

若くて，勇気や信念が人一倍ならなんでもできるとはかぎらない。勇気があっても若い人にはなかなかむずかしいことだってある。だが非凡な人もいるものだ。

M君は経済学部に籍をおく現役の学生で20歳である。

じつはこの連載で以前に，NIFTY-Serveの「サーチャー倶楽部」主催の朝日新聞社のデータベース見学会の折に，現場で説明を聞きながらリアルタイムでノートパソコンに打ち込んでいた，猛スピードのタイピングの主として登場した人でもある。

M君は学業のかたわら専門分野の勉強を生かして，すでに情報やコンピュータ関連の実務にたずさわっているらしく，広い知識や経験をおもちのようだ。ほんとはM君なんて呼ぶのはとても気がひける。

その彼があることを発端にひとつの追究をした話が，時間を追って「サーチャー倶楽部」で語られていた。

M君はある大きな会場でおこなわれたメディアのイベントで，アンケート調査に応じて自分に関するデータを回答した。そのあと，「新しい種類の情報サービスを提供する会社」の営業マンと称する人から電話でのアプローチがあり，数日後，自宅近くの喫茶店で面会した。

N社というその会社の「Nクラブ」の会員になると，「Nデスク」の利用で内外のあらゆる情報サービスを受けられる。また英会話教室もある。「Nデスク」はサーチャー6人が常駐してサービスにあたり，情報図書館や専門家集団との提携もある。コンピュータ通信，FAX，電話，ボイスメール，郵便などのメディアにより，万全の対応ができる。この会員になるためにはクレジット会社経由で80万円ほど支払うのだが，いまこの場で申し込むとFAX(5万円相当)と英会話教室（20万円相当）が無料になる。

英会話などは学生を対象にしたセールスとしてよく聞く話だが，情報に関するサービスは，そのしくみを知る人にとっては斬新で興味のもてるものだと思う。

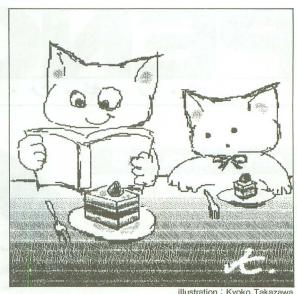
illustration：Kyoko Takazawa

M君はよく考えて決めたいので2日待ってもらいたい，FAXと英会話の無料サービスもそれまで待ってもらえないかと聞いてみたら，断固ダメという返事だった。けっきょく，契約せずに相手と別れた。説明を聞くために3時間かかったそうだが，3時間もセールスマンと2人だけですごしながら，最後に自分の判断をしっかりとくだしただけでもスゴイなと思っていたら，それだけではなかった。

社長との一問一答

M君は勧誘の方法や業務内容になにか割りきれないものを感じて，疑問点を「サーチャー倶楽部」に書き込んだ。それをN社の社長が目にした。じかに得たのか，あるいは誰かを介してか，公開されているネットの便利さとスリルだ。社長はM君に面会をもとめてきた。

そしてM君は社長のまねきで，もちろんひとりで，新宿の三井ビル内のオフィスに彼をたずねた。そこであらためて，N社のなりたちや業務についての説明をくわしく受けた。N社は名高い精密機器メーカーが参入している会社グループの一員であること。親会社にあたるA社は国際交流事業をはじめ，教育，出版，留学など手広い事業を手がけていること。それからN社の業務とNデスクについて。

M君は面会以前に自分の見解や質問すべき事項を準備，研究しておいた。東京都消費者センターが発行している雑誌の特集記事「若者の消費者トラブル」から，実在のある会社が，法律や条例に違反の疑いがある問題点としてあげられている項目を例に，ひとつずつ社長に質問した。

「電話によるアポイントメントセールス」で，本来の目的を隠した営業をしていないか。「特典ばかりを強調した説明」がされていないか。「長時間の勧誘で，疲れきって冷静な判断もできない」状態をつくっていないか。「親にはいわず，支払いは自分ひとりでするように」といった強引な契約方法をとっていないかなど。M君は営業マンからじっさいに，「自分への投資だから親には説明しないで，自分で決断し自分自身で支払うよう」いわれたそうだ。

ほかにM君は会員数や会員権の転売，情報料などについて，質問をした。

社長の回答は要を得ないものも多かったが，セミナールームや商品展示室を見学させてもらったり，別の場所にあるオープン予定のインテリジェントビルなども見せてもらった。M君としては，営業マンの勧誘方法にまだ疑念はあるものの，自分の目で一部をたしかめることができて，なっとくのいくものもあったらしい。

社長のほうでも，今後のイベント会場などでの勧誘方法には改善のつもりがある旨をM君につげたそうだ。

47歳だという経営の責任者と，ここでも3時間以上の会見をもって，しかも自分が用意した疑問点を臆せずに逐一たずねてみる。計画だけはできても実行できる人がどのくらいいるだろうか。周到な準備と一貫した冷静さで，目的をきちんと達成できる人。これは勇気だけの力ではない。

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

ハードディスクドライブユニット
MK-HD2
満開製作所

MK-HD2

満開製作所はハードディスクドライブユニット「MK-HD2」(満開式硬盤駆動装置弐號）を発売した。

本機は東芝製1.0Gバイトハードディスク1基を内蔵したSCSIハードディスクユニットである。キャッシュバッファは512Kバイトを内蔵していて，平均アクセスタイムは8.33msとなっている（同社調べ）。

フル↔フル，フル↔ハーフ対応ケーブルが同梱されている。大きさは，250mm(幅)×290mm(奥行)×75mm(高さ)で，重さが3.8kgとなっている。

価格は150,000円のところ，現在は特価125,000円（税別）で販売中。

〈問い合わせ先〉
㈱満開製作所　☎03(3554)7441

プリンタ
MJ-500V2/1000V2/AP-1000V2
エプソン

エプソンは，マッハジェットプリンタ2機種「MJ-500V2」，「MJ-1000V2」とハイパーサーマルプリンタ「AP-1000V2」を発売した。

「MJ-500V2」は同社が開発した積層ピエゾヘッド“MACH”によって，用紙に吹き付けるインク滴の形状を均一化してくっきりとした印字を実現した。縮小モードとして80%と50%の2種類を標準装備している。

MJ-500V2

MJ-1000V2

AP-1000V2

A4用紙1枚あたりの印刷コストは約3.2円。オートシートフィーダを標準装備して，B5/A4用紙，最高100枚の連続給紙が可能になった。さらに，手差し給紙も可能。

「MJ-1000V2」は「MJ-500V2」の機能に加え，オートシートフィーダはA3サイズまで対応している。ほかにも，連続紙対応のトラクタユニットを標準装備。マルチウェイローディング機構により単票紙とワンタッチで切り替えが可能になっている。また，オプションのインタフェイスカードを追加することで，最大2種類のパソコンを同時に接続することが可能。

「AP-1000V2」は，8色のカラー印刷に対応している。A4用紙1枚あたりのコストは約4円。縮小拡大機能は50%，80%，125%の3種類がある。オプションのカットシートフィーダはハガキで30枚，A4で50枚の連続印刷を可能にする。

この3機種はプリンタ制御コード「ESC/P」をバージョンアップした「ESC/P V.2」を搭載している。それによって，アウトラインフォント（明朝体，ゴシック体，毛筆体を標準装備）での印字を可能にした。

価格は，「MJ-500V2」が69,800円，「MJ-1000V2」が99,800円，「AP-1000V2」が69,800円（それぞれ税別）となっている。

〈問い合わせ先〉
エプソンインフォメーションセンター
☎03(3377)3500，06(212)8715

OS-9/X68030対応ウィンドウシステム
X Windows V11.5
マイクロウェアシステムズ

X Windows V11.5

マイクロウェアシステムズはOS-9/X68030対応のウィンドウシステム「X Windows V11.5」を発売した。

このソフトは米国マサチューセッツ工科大学が開発した「X Window System」のクライアントおよびサーバをOS-9/X68030に移植したものである。

特徴としては，「X Version11 Release 5」を完全に満たしている。また，OS-9をスタンドアロンで使用しているときのためにクライアントとサーバをサポートした。Xlib，Xaw，Xt，Xmuなどのクライアント開発環境用のライブラリが用意されている。ほかにもMotifウィンドウマネージャ，ISP

V1.4.1，インストレーションマニュアルが同梱されている。動作に関しては，メモリが最低8Mバイト（10～12Mバイト推奨），ハードディスクの空き容量が80Mバイト以上必要となる。

価格は30,000円（税別）。

〈問い合わせ先〉
マイクロウェアシステムズ㈱　☎03(3257)9000

パーソナルワープロ
WD-A850/A950
シャープ

WD-A850
シャープはパーソナルワープロ２機種「WD-A850」「WD-A950」を発売した。

「WD-A850」は書院スーパーレイアウト機能により，画面上で実際の印刷イメージを確認しながら文書を作成できる。ペン機能では，手書き文字入力のほかに罫線を引いたり編集作業などが行えるようになった。印刷は，印刷位置を示すLEDの光る指標を動かすことによって，位置合わせを行う。書体も書院スーパーアウトラインフォント６書体を搭載している。ほかにも，光通信コードレスの10キーステーションを標準装備している。

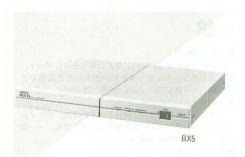
「WD-A950」は「WD-A850」の機能に加え，報告書や各種届出書など，枠組みされた定型用紙への印刷が可能。自分で作成したフォーマットの保存もできる。

価格は「WD-A850」が250,000円，「WD-A950」が270,000円（ともに税別）。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221,043(299)8210

超薄型無停電電源装置
BX5
オムロン

オムロンは超薄型無停電電源装置「BX5」を発売した。

本機は回路部とバッテリー部を別にした２ボックスタイプの無停電電源装置である。９月に発売された「BX3」の出力容量を500VAに増やしたものでパソコン２台を５分間バックアップすることができる。

BX5

バッテリーの交換は，別売りのバッテリーユニットを購入し個人で行うことが可能。また，ブザーとLED表示により，停電，過負荷，バッテリーローの状態を知らせる。

大きさは「BX3」と同じで190mm（幅）×335mm（奥行）×41mm（高さ），重さは６kgとなっている。設置方法も縦置き，横置き，重ね置きなど使用環境に応じて選べる。

価格は47,000円（税別）。

〈問い合わせ先〉
オムロン㈱　☎03(5488)3221

高機能VRマウス
IVR-68
スピタル産業

IVR-68

スピタル産業は高機能VRマウス「IVR-68」を発売した。

本機はマウス上部のセレクトボタンを押すことによって５段階（100，200，300，400，AUTO）に分解能の切り替えが可能。各モードはLEDインジケータによって確認できる。AUTO機能はマウスを動かす速度によって自動的に分解能が４段階に変化していく。ターボ機能はマウス左側面のターボボタンを押しているあいだ，どのモードでも強制的に800dpiモードにしてしまう。ほかにも，ファジーリバイス機能により，移動方向と反対側に動かそうとする外乱に対して自動補正を行うこともできる。裏面のスイッチで切り替えが可能。

価格は9,800円（税別）。

〈問い合わせ先〉
スピタル産業㈱　☎03(3251)2918

GCCによるX680x0ゲームプログラミング
ソフトバンク

ソフトバンクはX68k Programming Series「GCCによるX680x0ゲームプログラミング」を発売した。

本書は，「C Magazine」に連載された「X68k活用講座 GCCによるゲームプログラミング」をもとにして加筆修正されたものである。内容は，コンピュータの基礎知識からC言語の入門などで，最終的にC言語で１本のゲームプログラミングを作成する。付録ディスクが２枚付属。なかにはゲームの全ソースプログラム，GCCの実行環境(GCC，HAS，HLK，GDB，LIBC）が収録されている。

価格は3,600円（税込）。

〈問い合わせ先〉
ソフトバンク㈱　☎03(5642)8101

INFORMATION

ストリートファイターIIダッシュ バトルロイヤル
満開製作所

12月26日に「ストリートファイターIIダッシュバトルロイヤル」が満開製作所主催，シャープ・カプコン協賛で開催される。

会場では，「ストリートファイターIIダッシュ」の勝ち抜き戦が行われるほか，宍戸留美ミニコンサートやX68000グッズ販売が予定されている。

場所は，シャープ市ヶ谷ビル8Fエルムホール，開催時間は11：30～17：00，コンサートの開始予定は12：00。入場は無料。

〈問い合わせ先〉
㈱満開製作所　☎03(3554)7441

FILES Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。いよいよ1994年に突入しました。受験生の方は最後の追い込み，がんばってください。あと，がんばりすぎて寝過ごさないようにね。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
テクノポリス　徳間書店
電撃王　主婦の友社
POPCOM　小学館
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶THE NEWS FILE
「Silicon Graphics EXPO'93」などの展示会のレポート，富士通ゼネラルから出た薄さ6cmのテレビの話題などを紹介。――編集部，LOGIN，22号，28-34pp.

▶テクノストレスの恐怖
コンピュータを使う人に現れる特有の症状，テクノストレスについて。症状や撃退法などを紹介する。――編集部，LOGIN，22号，246-249pp.

▶MUSIC LABO
MIDI音源を取り上げる。最新の音源6種類を紹介し，他に立体音響のトピックなどを掲載。――編集部，LOGIN，22号，250-253pp.

▶電網幼稚園
パソコン通信をより快適に利用するための講座。今回は電話料金をいかにして節約するかを説明する。――編集部，LOGIN，22号，278-279pp.

▶特捜情報最前線2
CD-ROM，AVスピーカー，サングラス形態のテレビなど，マルチメディア時代への足取りを感じさせるグッズの紹介。――編集部，コンプティーク，12月号，32-33pp.

▶TOKYO GAMERS' STYLE!
任天堂との提携で一躍脚光を浴びたシリコングラフィックス社の紹介から，映画，OVA情報まで。――編集部，コンプティーク，12月号，179-194pp.

▶巻頭特集①　ゲームギョーカイ　噂と真相
任天堂をはじめとしたコンシューマー向け新機種の情報。年末年始の状況から推察する来年のパソコンゲームの展望など，業界のトピックスを追いかける。――編集部，電撃王，12月号，6-15pp.

▶巻頭特集②　アーケードゲーム大特集
「サムライスピリッツ」と「餓狼伝説SPECIAL」を取り上げて攻略法を研究。そのほか新作アーケードゲーム情報もあり。――編集部，電撃王，12月号，16-23pp.

▶特集　日本のテーブルトークRPG10年史
過去10年間に登場したテーブルトークRPGのなかから厳選した50本を紹介し，その変遷をたどる。――安田均，村川忍，電撃王，12月号，86-93pp.

▶徹底検証・美少女戦士セーラームーン
幅広い層にファンをもつ「セーラームーン」を題材としたゲームやグッズを研究し，キャラクターゲームの未来を探る。――編集部，電撃王，12月号，158-161pp.

▶新鮮良品館
ミニコンポ特集のほか，ミニサイズのビデオ「ミデオ」やJBLのスピーカーシステムなど各方面のアイテムを紹介する。――編集部，POPCOM，12月号，126-127pp.

▶HDDハードディスク
HDのしくみを解説し，初めて買う人のために製品購入のアドバイスをする。HD現行機種一覧つき。――編集部，マイコンBASIC Magazine，12月号，43-57pp.

▶Bug太郎のプログラム・タイム
ゲーム作成などで役に立つ「放物線を描くアルゴリズム」と「誘導弾のアルゴリズム」を解説する。――編集部，マイコンBASIC Magazine，12月号，88-89pp.

▶0から始めるマルチメディア
現在のマルチメディアの状況をさまざまなジャンルから検証する。――編集部ほか，I/O，12月号，9-33pp.

▶COMPUTER CURRENTS
現在もっとも成長著しいソフトハウス，マイクロソフト。その功績と今後のビジョンを考える。――Tim Bajarin，I/O，12月号，59-61pp.

▶Silicon Graphics EXPO'93
横浜アリーナで10月13日から2日間開催された「Silicon Graphics EXPO'93」をレポート。――編集部，I/O，12月号，73-75pp.

▶スーパーコンピューティング入門　No.36
「カオスとフラクタル」をテーマにしたシリーズ第11回。今度はフラクタルの華，マンデルブロー集合とジュリア集合について解説する。――林智雄，I/O，12月号，143-146pp.

▶特集1　最新機種・うちはマシンが新しい
新製品のテストレポートと，現在の各社主力機種一覧およびマシン選びのガイドライン。MacintoshとFM TOWNSの新ラインナップも詳しく報告する。――編集部，ASCII，12月号，285-308pp.

▶PRODUCTS SHOWCASE
16万円以下のページプリンタの新製品，スキャンコンバータ3種類，バブルジェットプリンタ「BJ-15v Pro」，緑電子の昇華型カラープリンタ「PRN-11」などのレポート。――編集部，ASCII，12月号，373-388pp.

▶極楽辞書引き計画　その拾
ソニーの電子ブックプレーヤー「DD-30DBZ」の詳細をレポート。――志村拓，ASCII，12月号，397-404pp.

▶TBN SPECIAL DIGI-VIS TODAY
デジタル映像とは切り離せないテーマのひとつが「圧縮」である。圧縮の必要性と圧縮の現状などについてジー・シー・テクノロジーの藤原洋氏に聞く。――編集部，ASCII，12月号，446-447pp.

▶バカパパのモノを買い物
「ウェットティッシュでなんでも拭え」がテーマ。各社の各用途向けウェットティッシュを特集。――編集部，ASCII，12月号，448-449pp.

▶特集　各種プリンタ選択のポイント
現在のプリンタ事情を印字方式別に解説し，最新の製品群をチェックする。――編集部，My Computer Magazine，12月号，27-42pp.

▶パソコンの快適環境をつくる
ハードディスクを使ううえでの基礎知識を伝授。カタログの読み方，専門用語の意味など。――佐田守弘，My Computer Magazine，12月号，74-77pp.

▶未来派パソコン通信の研究〈2〉
現在の通信方法とその問題点に迫る連載。今回は電子メールの集中管理方式について。――原田洋平，My Computer Magazine，12月号，142-143pp.

▶全国的電脳地図虎之巻
秋葉原，日本橋をはじめ全国8エリアのパソコンショップを紹介する特別付録。X68000に強い店の紹介もある。――編集部，LOGIN，23号，別冊

▶THE NEWS FILE
「PC-486PORTABLE」登場のニュースや，新鋭Macintoshの紹介，デジタルスチルカメラ初体験のレポートなど。――編集部，LOGIN，23号，32-39pp.

▶特集　新世代ゲームをプレイせよ！
最新のハードでないと動かない海外からの超大型ゲームが登場している。それを楽しむための環境を考える。――編集部，LOGIN，23号，227-241pp.

▶電網幼稚園
パソコン通信生活を送るなかで気になりだすのが，膨大な量のログファイル。今回はこのスマートな解決方法を伝授する。――編集部，LOGIN，23号，280-281pp.

X1/turbo/Z

X1シリーズ

▶NUMBERS
2人対戦落ちものパズル。横一列に同じ数字をそろえて相手を上に押し上げるのが目的。――SACHI SOFT，マイコンBASIC Magazine，12月号，136-137pp.

▶FREEZE／2
倉庫番風のパズルゲーム。ブロックを押して道を作り，右上のゲージいっぱいにクリスタルを集める。1993年7月号に掲載されたゲームの改良版である。――森敬雄，マイコンBASIC Magazine，12月号，138-139pp.

▶スピンディジーII～Eternal Move～
ミュージックプログラム。――RUFINA，マイコンBASIC Magazine，12月号，156-157pp.

X68000

▶最新ゲーム徹底解剖!!
光栄の「項劉記」など，各社最新ゲームの攻略解説記事。懐かしGAME REVIEWには「うっでぃ・ぽこ」が登場。――編集部，LOGIN，22号，116-169pp.

▶特集　特撰！ ニューソフトキング
年末年始のニューゲーム情報を集めて一挙放出。X68000用「卒業」「ストリートファイターIIダッシュ」な

ど。――編集部，LOGIN，22号，203-245pp.
▶X68030新聞
ついに登場，「ストリートファイターIIダッシュ」ほか，近日発売のゲーム情報。SX-WINDOW開発キットとCコンパイラの情報も掲載。――編集部，LOGIN，22号，258-259pp.
▶Hot Press SPECIAL
「ストリートファイターIIダッシュ」を紹介。開発進行状況の報告つき。――編集部，POPCOM，12月号，22p.
▶HOT PRESS ＋1
各ソフトハウスから発売されたゲームを紹介。「ぷたさん」「宝魔ハンターライム5」「麻雀クエスト」ほか。――編集部，POPCOM，12月号，32-35pp.
▶SUPER SOFT EXPRESS
「ストリートファイターIIダッシュ」を紹介。――編集部，コンプティーク，12月号，45p.
▶How to Win
X68000版「項劉記」「信長の野望・覇王伝」など新作ソフトの攻略法を研究する。――編集部，コンプティーク，12月号，94-97，102-105pp.
▶Dengekiパソコン
新作チェックと，年末発売予定のゲームを特集。「ネメシス'90改」を大きく取り上げている。――編集部，電撃王，12月号，52-70pp.
▶NEW GAME REPO!!
「ネメシス'90改」「宝魔ハンターライム6」などを紹介。発売予定ソフトの機種別カレンダーつき。――編集部，テクノポリス，12月号，43，48，54pp.
▶HOT REVIEW!!
新作ゲームソフトを業界有名人や読者が語るコーナー。今月はX68000用「コットン」など。――編集部，テクノポリス，12月号，58-66pp.
▶DO-JIN SOFT FAN!!
アマチュアソフトとサークル情報のコーナー。クイズゲーム「QUIZふろんてぃあ」，ポリゴン処理の3Dゲーム「ARTEMIS」などを紹介。――編集部，テクノポリス，12月号，67-73pp.
▶高校教師Sim Teacher
最近はやりの育てるゲーム。高校教師となって生徒の学力をアップさせるのだ！ ――西村志郎，マイコンBASIC Magazine，12月号，140-141pp.
▶TANK COMMANDER・X
1対1の戦車戦。戦車同士で撃ち合うシューティング＆シミュレーション。2人専用。――間部靖史，マイコンBASIC Magazine，12月号，142-144pp.
▶闇の血族「上巻」～オープニング～
ミュージックプログラム。――藤井亘，マイコンBASIC Magazine，12月号，158-159pp.
▶ベーマガ㊙情報局
「悪魔城ドラキュラ」「クレイジークライマー/クレイジークライマー2」などの秘密情報を紹介。――編集部，マイコンBASIC Magazine，12月号，222-223pp.
▶SUPERSOFT HOT INFORMATION
X68000用新作3本を紹介。「餓狼伝説2」など。――編集部，マイコンBASIC Magazine，12月号，別冊9p.
▶AV STRASSE
X68030対応の標準Cコンパイラ「C Compiler PRO-68K ver2.1 NEW KIT」，OS-9の開発環境「OS-9/X680x0 Ultra C&Professional Pack V1.1」などの内容を紹介する。――編集部，ASCII，12月号，413-416pp.
▶FREE SOFTWARE INDEX
大手主要ネットにアップロードされたソフトのなかから紹介する。X68000用汎用コピーツール，ファイラーなど。――編集部，ASCII，12月号，483-489pp.
▶CコンパイラPRO 68K Ver2.1 NEW KIT
X68030にも対応し，その性能を生かしたプログラミングが可能になるCコンパイラをレポートする。――高橋雄一，My Computer Magazine，12月号，161-163pp.
▶なんでもQ&A
「Easydraw SX-68K」のデータをシャーペン.X上にドローデータとしてペーストするにはどうしたらよいか，などの質問に答える。――シャープ株式会社，My Computer Magazine，12月号，170-171pp.
▶NEWSOFT
「ストリートファイターIIダッシュ」「ネメシス'90改」「宝魔ハンターライム」ほかを紹介する。――編集部，LOGIN，23号，12-31pp.
▶X68030新聞
年末の大作が目白押しのX68000。今回は「餓狼伝説2」「ドラゴンバスター」「ハイパーピクセルワークス」など。――編集部，LOGIN，23号，260-261pp.
▶SX-WINDOWプログラミング
XGCCでSX-WINDOWアプリケーションを開発するために必要なプログラム類を付録ディスクに収録。その内容について概略を説明する。――吉野智興，C Magazine，12月号，142-148pp.

ポケコン

PC-E500

▶KEYBOARD BATTLER
バーコードバトラーのポケコン版。バーコードのかわりに，適当にキーボードを叩いて戦士を作って戦わせる。――TOPPE，マイコンBASIC Magazine，12月号，146p.
▶CORORON2
落ちものアクションパズルゲーム。ブロックを消していく条件がちょっと変わっている。――近藤紀之，マイコンBASIC Magazine，12月号，147-149pp.

新刊書案内

超日常観察記
岡本信也＋岡本靖子著
情報センター出版局刊
☎03(3358)0231
A5判 221ページ
1,400円（税込）

考現学というと，どうしても頭に浮かぶのが赤瀬川原平氏らが中心になっている「路上観察学会」だが，実は「路上観察学会」より何年も前から，名古屋で活動していた考現学のグループがあった。本書「超日常観察記」は，そのグループのメンバーによる考現学のまとめだ。「路上観察学会」が文字どおり「路上」を中心にして，路上に何気なく発生した隠れた芸術を求める（例：トマソン）ものであったのに対し（まあ，メンバーがメンバーだから，普通の考現学で終わるはずがないのだが），こちらは，地道に，あらゆるものを「考現学」していく。定点観測を含むけっこう地道なフィールドワークの成果だ。
傑作なのは，「人の観察」である。これが凄い。おばあさんの履物の観察（老人を観察したくなる気持ちはよくわかる），野良着の観察，女性のバッグの観察，女性の下着の観察，銭湯やその客の観察（夫婦で男湯と女湯それぞれ観察している），さらには食前の仕草，定食の食べ方，乗客の視線，足の組み方にまで至るのだ。どれも，したり顔の現代社会考察ではなく，フィールドワークの結果が述べられているのである。それも，何気ない観察ではなく，しっかりとメモを取り，自分の目で調査した結果なのだ。面白くないわけがない。
もちろん，街に放置されたモノたちの観察や，残っている古い文化の観察など，フィールドワークの範囲は広い。考現学はディテールへのこだわりでもある。ただ「バケツ」があるのではなく，「バケツ」にもさまざまな顔があり，さまざまなくたびれ方をしているからだ。
考現学の面白さは，一見くだらないことを真面目に観察する面白さであり，発見の面白さであると同時に，ディテールへのこだわりの面白さにある。面白さを誇張して芸術にまで高めたのが「路上観察学会」であり，誇張することなく淡々と綴ったのが本書だと思えばいいだろう。 （K）

カオス
まったく新しい創造の波
合原一幸著
講談社刊
☎03(5395)3622
四六判 229ページ
1,800円（税込）

「一瞬の蝶の羽ばたきが未来を永劫に変えてしまう」このフレーズはカオスを語るときによく出てくる。ある時期，マスコミなどでも，やたらと取り上げられた。しかし，その本当の意味をどれだけの人が正確に理解していただろう？
本書はカオスについて，一般の人向けにわかりやすく解説している。特に，カオス理論が発見されるまでの歴史やカオス現象の実例，カオスの有用性の具体例などについて語られている。
カオスという言葉はよく聴いたけれども，意味がよくわからなかったという人は，それを真に理解するためにも読んでみてはどうだろうか。

複雑性の科学
コンプレクシティへの招待
ロジャー・リューイン著
糸川英夫監修
福田素子翻訳
沼田寛解説
徳間書店刊
☎03(3433)6231
四六判 322ページ
2,000円（税込）

コンプレクシティ（複雑さ）は新しい科学である。自然や生命，文明などすべては複雑に変化するという主張だが，そこに秩序を認める点で，非線形で無秩序な相違というカオス理論とは一線を画す。
規則性と不規則性がバランスを保ちながら，より複雑に発展していく，その臨界領域を，著者は「カオスの縁」と名づけた。そして，複雑さの根底にすべての現象を解き明かす鍵となる共通法則を見いだすべく，さまざまな分野の専門家たちに問いを投げかけていくのである。
ただひとつの真理となる鍵はあるのか，コンプレクシティはその探索への新たな道である。

QUESTION and ANSWER

X68000上で動くFORTRANを探しています。ありますか？

東京都　井戸　浩登

最近は使われることが少なくなったとはいえ，FORTRANの遺産（プログラムやASPPなどのさまざまなライブラリパッケージなど）は膨大なものがありますし，多くの大学のプログラミング実習などでは，いまだにFORTRAN77が使われていたりもします。確か清水和人氏などは仕事ではFORTRAN以外使わないといっていましたし，使われているところではFORTRAN処理系は依然としてメジャー言語として猛威をふるっているようです。

いきなり結論ですが，現在のところ，Human68k上で動作するFORTRANコンパイラはありません。

以前フリーソフトウェアのFORTRANを探したこともあったのですが，出回っている処理系自体がほとんどありません。やっとみつけたAMIGA用のFORTRAN77コンパイラも，ソースの入手法がわからなくてどうしようもありませんでした。ソースつきで配布されているものとなるとCP/M時代のもの（かなりサブセット）しかないのかもしれません。

用途にもよりますが，すでにあるFORTRANプログラムを持ってきたいというのであれば，GCC関連ツールのF2Cを使用されることをおすすめします。これはFORTRAN用のプログラムをC言語用に変換するツールです。ちにみにこのツールは『X68000Develop』（いわゆるGCC本）にも収録されています。

自分でプログラムを作成するというのであれば，コンバータよりはちゃんとしたコンパイラがあったほうが望ましいのはいうまでもありません。

入手が非常に困難であったり媒体の問題も多々あるとは思いますが，CP/M68KやOS-9/68000関連ではFORTRAN処理系がリリースされているはずですのでそのへんをあたるという手もあります。こういった言語は汎用で作られているのでX68000上のCP/M68K（あるいはCP/M68Kエミュレータ）やOS-9/68000上でもおそらくちゃんと動作するものと思われます。

やはり，ちゃんとしたFORTRAN処理系（できればFORTRAN90）というのもほしいものですね。

SCSIのハードディスクを持っているのですが，最近不連続点が多くなってきたような気がします。どうしたらよいのでしょうか。ちなみに，僕は通信をやっていないので“REFRESH.X”というのはほぼ入手不可能です。一度ハードディスクをフォーマットしてリストアしても無駄ですよね？

京都府　村上　浩二

これはハードディスクの宿命とでもいうべき問題です。ハードディスクは高速なメディアですが，厳密な意味ではランダムアクセスをしていませんから，ファイルの記録状態によって読み書きの速度にバラツキが出てきます。

データ読み込み時のハードディスクの動作は2つに分けられます。すなわち，データがある位置まで磁気ヘッドを動かすシーク動作と実際にデータを読み込む読み込み動作です。

100Kバイトのファイルがあったとして，これが連続領域に格納されている場合には，1回のシークと1回の読み込みですみます。しかし，これが10カ所にわたって不連続になっていると，シークが10回，読み込みが10回になります。

シーク速度は実際にディスクに記録されている位置と磁気ヘッドとの距離によって変わってきます（外側と内側の物理的距離)。いくら高速ディスクといっても最内周から最外周へ磁気ヘッドを動かすには時間がかかります。また，連続領域の読み込みは1回の読み込みサイズが大きいほど効率よく読めます。

よって，OSはできるだけファイルを連続領域に書き込もうとしますが，それが物理的に不可能な場合はファイルがいくつかの領域に分散して記録されることになります。これが不連続点となります。

どのような場合に不連続点ができるかというと，ディスク容量がいっぱいに近くなったときに特に発生しやすくなります。大きさの違うファイルを何度も書き込んだり消したりしていると，だんだんと隙間が増えていき，どうしてもそれらをかき集めなければならないような状況になるからです。こうなると使っていてもファイルの読み書きに重さを感じるようになります。最悪の場合，フロッピーディスクより遅くなることすらあります。

REFRESH.X（SCSIハードディスク用にはREFRESHG.X）というのはこういった不連続領域がなくなるようにファイルを並べ直すというツールです。最近では通信をしていなくてもフリーソフトウェア集などで入手できますが，ディスクを直接書き換えるものだけに，使用する際には必ずバックアップをとってから行うようにしましょう。

こういった特殊なツールを使わずに，もっとも確実に対処する方法は「まっさらなメディアにファイルを順番に書いていく」というものです。

村上さんのいうとおり，BACKUP.X/RESTORE.Xなどを使うとディスクの情報をベタで持ってきて，ベタで格納/復帰しますのでこういった用途にはまったく向いていません。

よって通常こういった目的には，ファイル単位でのバックアップと復帰を繰り返すCOPYALL.Xを使用します。

まず最初にハードディスクをCOPYALL.Xでバックアップしてください。途中でトラブルがあった場合，復帰が困難になりますので，念のためにフロッピーディスクでのシステムディスクを用意しておきましょう。

続いて，ハードディスクをフォーマットします。これは領域解放と領域確保だけでかまいませんが，ついでに装置初期化をしておくのも悪くありません。

あとはまたCOPYALL.Xでハードディスクにファイルを戻してやるだけです。領域解放と領域確保を連続して行ったときに，確保したパーティションが同じ容量であった場合リセットしなくてもハードディスクに書き込みを行うことができますが，ここでは必ずリセットしてから使用するようにしてください。一見，動いているようでもあとあとトラブルの原因になることがあります。

ハードディスクは残り記憶容量にある程度余裕を持って使うというのが正しい使い方です。バックアップを取る前にハードディスクの内容は十分に整理しておきましょう。そうすることで無駄なファイルまでバックアップすることを防ぐことができ，作業効率も上がります。

Z-MUSICで音楽が鳴っている最中にBASICのコマンドA_PLAY()で音を出してもドラムでかき消されてしまいます。かき消されないようにする方法はないですか。

静岡県　杉浦　詳和

Z-MUSICシステムではFM音源やAD PCM，MIDIの演奏データを効果音として，現在演奏しているデータに割り込むように演奏指定することができます。いわゆる効果音モードが用意されているのですが，残念ながらX-BASIC用にはサポートされていません。FM音源を使った効果音やMIDIを使った効果音を割り込ませることはできませんが，この質問の状況のようにAD PCMの効果音に限ればZ-MUSIC専用のX-BASIC外部関数MUSICZ.FNCで用意されているM_PCMPLAY()という関数が使えます。単にAD PCMの効果音を使いたいというだけならこれで十分だと思います。

これは，

M_PCMPLAY(n, p, f)

のように使用します。nは鳴らしたいAD PCM音が登録されているノートナンバー（0～511），pはパンポット（0～3），fは再生周波数（0～4）となっています（パンポットと再生周波数はA_PLAY()と同じ仕様です）。

あらかじめ，曲データで使用するAD PCMのAD PCMコンフィギュレーションファイル中に使いたい効果音も一緒に登録しておき，そこで指定したノートナンバーの効果音をBASICプログラム中のM_PCMPLAY () で使用してください。もちろん，MUSICZ.FNCの登録を忘れずに（付録ディスクに収録のバージョンでかまいません）。

なお，FM音源やMIDIの効果音が使いたいという場合はあらかじめ空きチャンネルを作っておいて，効果音ごとに専用のトラックを用意してそれを途中で演奏してやるようにします。

X-BASICでは同じチャンネルを使った割り込みはできませんが，実用上さほど問題にはならないでしょう。これらはC言語用ライブラリーZMUSIC.Lでもサポートされていますので X-BASICで作成したプログラムをコンパイルしても支障はありません。

カウンタ表示のよくわからない友達がいます。あまり詳しくないのでくわーしく教えてください。

福岡県　帆足　徹也

おそらくLIVE inで使用されているZ-MUSICのカウンタ表示のことだと思われますので，それについて解説します。

これは簡単にいえば演奏データ中の各トラックで使用している音長値をすべて合計したものです。Z-MUSICの基準クロックで計ったものを16進数で表示します。

最初の数字がトラック番号で，次に総ステップ数，ループ内部でのステップ数が表示されます。

これはZMUSIC.Xの持っているチェック機能のひとつで，

ZMUSIC-Q TEST.ZMS

のようにすることでTEST.ZMSのカウンタ表示を行います。

入力したミュージックデータのどこかに誤りがあった場合のうち，音長に関する間違いを検出するためのものです。音長値の表記はMML中での省略も多く，チェックが難しいこと，Z-MUSICでは繰り返しなどの制御構造が複雑なことなどから設けられたものです。特にループ内での音長間違いは全体の演奏に大きな影響を与えるため独立に表示するようになっています。

とりあえず，LIVE inに掲載されているミュージックデータを打ち込んだ場合，

ZMUSIC -CTEST.ZMS

のようにして文法チェックを行い，続いて，上記の方法でカウンタ表示を行って数値がちゃんとあっているかどうかを確認するようにしてください。

ちなみに，このカウンタ表示がちゃんとあっていても完全に入力が正しいという保証にはなりませんが，この数値があっていないようだと確実に変な演奏になります。

ストリートファイターIIダッシュを買ったのですが，マニュアルに記載されているとおりにしてもハードディスクへのインストールができません。もしかしたら，ハードディスクの最初のパーティションでないと受け付けてくれないのでしょうか？　以前なにかでそのようなアプリケーションがあったように思います。ちなみに私の環境ではHuman68kの書き換え時のために，最初のパーティションにはシステムファイルしか入れないようにしているのですが（要するに1Mバイト分しか領域確保していない）。

山口県　亀山　博文

はっきりいってしまえば，これはマニュアルの記載ミスです。ストリートファイターIIダッシュのマニュアルに記載されているインストールの方法のうち，

MD SF2

にあたる部分を，

MD　STREET_FIGHER2

に修正して入力すればマニュアルの記述どおりに起動できるようになります。

SX-WINDOWなどを使用すればディレクトリのリネームもできますので，試してみてください。

また，普通にシステムを起動してストリートファイターIIダッシュが格納されているディレクトリに移動し，

AUTOEXEC

と入力することでもゲームは起動できます（ドライブ0にストリートファイターIIダッシュのシステムディスクを入れておけば)。なにかの作業をしていてふいにゲームがしたくなったときなどはこのほうが簡単でしょう。バッチファイルを見るかぎり，ゲームを終了してまたシステムに戻ることもできそうなのですが，残念ながらこれはできないようです。　（中野　修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

新年明けましておめでとうございます。月日に関守なしというように，あっというまに1994年。昨日はすでに今日の昔，でも明日はまだ手つかずだから，今年はがんばるぞ！　まずはコタツに入ってゆっくりミカンでも食べながら……。

◆またまた未開の地へと私を誘い込むOh!X。3D画像についてはまったくの素人の私ですが，これを機に勉強しようと思います。

佐村　和亮(18)福岡県

がんばってください。新しく知ることはなかなか楽しいものですよ。苦しみも多いけど……。

◆「SLASH」の記事を読んでいると，行間から皆さんの情熱があふれてくるようです。

臺　正史(21)埼玉県

今度は読者の皆さんからのあふれてくるような情熱にも期待します。

◆ああっ，2浪はしたくない。そうだ！「SLASH」を使いこなせるようになったら数学に強くなれるかもしれない。そうだろう，きっとそうだろう，そうだったらいいな。来年こそ……。

村松　充志(19)福岡県

思い込みが大切かもしれない。でも，数学だけで大丈夫ですか？

◆もうOh!X誌上で荻窪氏の記事を読むことはないのでしょうか。氏独特の語り口が気に入っていただけに残念です。たしかにX68000はビジネス，実用の面に弱すぎます。もう少しなんとかならないものでしょうか。

竹本　郁馬(22)千葉県

大丈夫です。これからも読めますよ。

◆Z'sSTAFF PRO-68K ver.2.0の明朝体は秀英明朝体というフォントで大日本印刷の書体です。たぶん著作権の問題で発売されなくなったのだと思います。PostScript用のフォントでしたら「海舟」という名前で明朝体とゴシック体が販売されています。原田　洋行(25)東京都

確認はとれていませんが，よく似ていますので，たぶんまちがいないと思います。

◆ジャンルのアンケートを見て思ったのですが，最近のゲームはジャンルが固定されすぎています。たとえば「格闘アクション」といえば「ストII」か「ファイナルファイト」のようなものしか想像できなくなっている状況ですよね。その昔の「ショーリンズロード」とか「スーパーチャイニーズ」みたいな毛色の変わったものもたまにはやりたいのに……。シューティングといえば「SHOT」と「BOMB」を使う縦スクロールしかありません。「サイバリオン」を初めて見たときのような新鮮な感動を与えてくれるゲームにもっと出合いたいです。

山下　智也(23)大阪府

いま売れているものに，そのジャンルのものが似てしまうのは，利益を考えるうえである程度仕方のないことです。でも，やっぱりいろんなゲームで遊びたいですよね。

◆11月号の97ページの伊藤さん，RPGを制作しているとのことですね。いまは受験なので停止中ですが，僕も仲間と制作中です。未熟で，all X-BASICで作ろうとしていますが，アセンブラは必要かなと思い始めているところです。受験を無事に突破できれば，制作再開になりますので，お互いがんばりましょう。応援しています。実はまだ僕も完成したことがないんです。

間渕　繁紀(17)静岡県

なにかを完成させることはなかなか難しいものですね。期待してますよ。受験のほうもがんばってくださいね。

◆ひょっとして，成人式は試験でつぶれるのでは!?

山崎　拓人(19)三重県

成人式って，ほとんどが成人の日にあるんですよねえ。でもその日は休日だから……ご愁傷さまです。まあ成人式の1回や2回出なくてもいいじゃないですか。

◆我が家のマウスはとてもキレイである。洗いもののプロがいるのだ。そいつは「ファジー」という名前のアライグマである。まず，石鹸を持たせる。数秒後，手にアワがつく。ファジーがそのアワを洗い落とす前に，マウスのボールを渡す。ゴシゴシ……これがまたなんともいえずかわいい。で，水を出してすすがせてやる。そしてここがポイント！　キレイになったと思ったら，すばやく奪い取るのである。でないと，ボールにかじりついてしまうのだ。タイミングをまちがえると，「グー」とうなり声をあげて僕の指をかじる。我が右手には生キズがたえない……。

塩瀬　勇人(18)神奈川県

うーんとってもかわいい。洗っている姿が目に浮かぶようです。編集部でも1匹飼ってみたいですね。無理かなあ。

◆9～10月にかけて「当たり月」でした。なにせ「電脳倶楽部」に載ってしまう。Oh!Xのプレゼントもいただいてしまう。そして落ちると予想していた講義も当たってしまい，いまや「当たらない気がしない」感じです。

坂井　国彦(21)静岡県

人生の幸運をこんなところで使い果たしてしまうとはもったいない。あっ，ひとりごとですから気にしないでください。

◆11月号152ページの加藤さんへ。鹿児島の教習所では，ある女性が教官から「エンジンブレーキをかけなさい」といわれたとき，えっといま踏んでるのがブレーキとアクセル，横にあるのがサイドブレーキでエンジンブレーキは……。で，彼女が踏んだのは教官用のブレーキだったそうです。女性がサイドブレーキをまたいでブレーキを踏む姿は涙なしには語れません。この話は本当にあったんでしょうか？　教官は遠いものを見るように語ってくれました。

手嶋　和徹(21)鹿児島県

同乗していた教官もびっくりしたことでしょうね。それにしても，ずいぶん足の長い

◀岡村　直也　兵庫県
いつまでも苦難の続く弟に未来はあるのか？　和が訪れると同時に別の苦難（ネタがない）が襲いかかるのか。ところで風邪には気をつけてね。平

◀佐藤　貴是　神奈川県
妙にシリアスなぶたさんたち。ゲーム画面からはとっても想像できない。でも，シリアスにしててもホノボノしているところがぶたさんかな。

女性ですね。私にはとても足りない。

◆11月号の小林さんへ。100m競走のランナーの場合は36km/hになります。とても人間ワザとは思えませんね。ちなみに自動車，自転車，バイク,馬,人間のうちスタートして一番速いのは？　①人間!?（1.97秒）②自動車（2秒）③バイク（2.47秒）④馬（3.03秒）⑤自転車（3.10秒）……ただし10mまでならの記録です。ダッシュではヒトパワーもバカになりません。

迫田　賢一(40)大阪府

100m競走をしている人ってけっこうな風圧を受けてるんでしょうね。

◆あったらこわい！　「おはようからおやすみまで暮らしを見つめるライオン」

垣内　学(22)三重県

たまにだったらいいかもしれない。よくない，よくない。

◆68030という石から，C_6という石に貯金が化けました(エンゲージリングってやつね)。それもX68030，2台分とは……。いつになったら我が家にツインタワーが来ることやら。

杉浦　誠(25)神奈川県

これは，先月の教えに従って，いま借金でもして買うのがいちばんですよ。そのあとどうなるかは責任持てませんけど……。

◆会社の先輩に薦められ，X68030のユーザーとなってはや3カ月。Oh!Xの読者となってはや6カ月。いまだにどちらも使いこなせてません。先輩は「Oh!Xは将来きっと役に立つから，いまはわからなくても買い続けるんだよ」といいますが，そんな日が来るのはいつのことやら。ちょっと不安です。だって難しいんですもん。

森　美和子(23)大阪府

いや～，わからなくても買ってくださるとはありがたいことです。わかる日が来るといいですね。それに応えるためにもがんばらなければ……。

◆11月号の「ショートプロぱーてぃ」の前置きの文章に一言。亜美ちゃんよりまことちゃんですよ。それにスクルドよりめぐみちゃんがいいですよ。……大きく道を踏み外した自分を感じます。水でもかぶって反省しよう。あ，これはマーキュリーでしたね，ハイ。

津村　忠蔵(18)佐賀県

なんのことだか私にはさっぱりわかりません。まあ，道がなくなったわけではないですから，いつでも戻れますよ……。

◆受験生らしく化学のお勉強をしていたら脳ミソがぶっとびました。$[FE(NH_4)_2(SO_4)_2\cdot 6H_2O]$（硫酸アンモニウム鉄（II）六水和物）なんてのがでてきたんですよ。昨夜は，ほぼ徹夜でチャットしていたし（本当に受験生か？）……。9月にモデムを買ったらみごとにはまりました。チャットはおもしろいけどみかか代が……。でも，パソ通はやめられない。現在，我が愛機は16MHzに改造中です。津田　将弘(18)愛知県

徹夜なんてしないほうがいいですよ。本当に……，はあ。

◆「満開製作所ってなんのメーカー？」と友人。「車にたとえると，トヨタのTRD，ニッサンのNISMOのシャープ版だな」と俺。「ふーん，じゃ車検通るんだ」と友人。すみません，車検通るんですか？　祝社長！　佐藤　仁(24)静岡県

6カ月車検なら大丈夫ですよ。本当か？

◆クロード・チアリがネットのホストをしているというのをテレビで見て驚いた。知らなかったのは僕だけ？　垣内　聡(18)和歌山県

私も知りませんでした。でも，いろんな有名人の方がネットを開いていたりするみたいですよ。

◆Oh!Xを読み始めて1年ほどですが，いまだに「LIVE in」などのプログラムの入力の方法がわかりません。一度初心者向けに誌上のプログラムの入力方法を教えてください。

井上　一哉(25)兵庫県

「ショートプロ」などでは入力方法について少しはふれていますが，ほかのものについてはあまりふれる機会がありません。今回の特集でわかっていただけるとありがたいのですが。

◆「アクエリアスMAX」はポカリスエットとポストウォーターを等量混ぜ合わせた液にラムネ菓子の粉末，チョコレート（胃が悪くなる安いやつ),隠し味にリアルゴールドまたはミラクルボディを入れたような味がする。それでも「メッコール」にはかなわない。うひ。

坪田　雅己(18)広島県

そんなに複雑な味でしたか。飲めないほどひどくはないような気がするのですが。でも「ミラクルボディ」ってなんですか？

◆事故りました。いや，自転車でなんですけど，やっぱり車とは恐ろしいものだとあらためて思いました。みなさんもお気をつけください（こんなの私だけだって)。　今井　佑(16)東京都

あなただけじゃありませんよ。特に広島の街を走っていると……。広島の方ごめんなさい，だって本当なんだもん。

◆巷で某法則が蔓延している。そのまねごとが書かれているのもよく見かける。でもそろそろイヤになってきた。皆さん，もうこれ以上日常を法則化するのはやめましょう。ノイローゼ気味な私です。「私は大きなあやまちはたまに犯すが，小さなあやまちはこれをよく犯す」法則化してるなあ……。　堀井　晶司(21)北海道

そのうち自分の行動をすべて法則化したりして……。

◆11月号の新井さんは，希望を持って公務員の仲間入りをされるそうですが（おめでとうございます),公務員ってまじめにやってきた人間がバカを見る世界だったりするときもあるところです。なぜか？　好況のときはボーナスはあまりに少ないし，不況のときはボーナスを削られるし……。ああ，X68030が遠ざかる～。でも首にならないのは最大のメリットかも!?

町田　華龍(27)神奈川県

公務員って首にならないんですか？

◆1990年2月号から買い始めたOh!Xを最初から読んでいると1992年9月号の響子さんのCGにリアルないかにもレイトレという感じの虫が描いてあった……と思ったら押し花のように潰れていたゴキブリの幼生だった。臭かった。バックナンバー買おうかな。しかしバックナンバーあるのかな？　片原　祥智(20)福岡県

いまこれを書いている段階ではバックナンバーはあるみたいです。それにしても見たくないなあ，潰れた姿は……。とりあえず合掌。

◆私がパソコンをいじりだしたのは，ほんの数カ月前からです。来月結婚する相手の部屋でホコリをかぶったゲーム機と化していたX68000を「ただのハコのままではもったいない！」と思ったのがきっかけでした。しかし持ち主であるダンナは「友達がパソコンを買い換えるときにゆずってもらっただけで，ゲーム以外には使ったことがない」という。会社にいるプログラマの人たちは「うちは98だからねえ」といい，妹までもが「うちも98だよ。ソフト多いもん。BASIC？　そんなの知るわけないじゃん。パソコンはゲームができればいーんだーい」ときた。そんな，数十万もするゲーム機にホコリをかぶらせるくらいならNEO・GEO買うわよ！　かくして私とX68000の戦いの日々は始まったのです。私の頼りはOh!Xだけよ。

阿部　一恵(25)神奈川県

ホコリのなかからよみがえったX68000。大

◀相沢　栄樹　東京都
「X-MAN参上」というタイトルだそうですが，まぬけなかんじがとってもいいですね。右上の可愛いキャラクターに名前はついてないんですか？

◀相沢　栄樹　東京都
「R.C.」で自分の作ったロボットのイメージだそうです。現在，最強を目指して改良中とのこと。勝負してみたいなあ。そんなに強くないけど。

切にしてあげてくださいね。難しい内容も多いと思いますが，まずはショートプロのX-BASICだけで動くものなどを試されてはいかがでしょうか？

◆初めて「チキンラーメン」を食べたがこれがうまい。カップラーメンとして正しくうまい。子供のころ食べた「おかしメン」を思い出してしまった。　中島　民哉(23)埼玉県

カップラーメンか。たまに食べるんならいいんですけど……。

◆もう，ため息しかでない……。Jリーグからサッカーを見始めた人は来年のW杯を絶対に見てください。日本のいないW杯は，これが最後だと信じて。　三浦　貴至(22)埼玉県

本当に残念でした。でもW杯は楽しみです。でも2002年のW杯の誘致は難しくなりましたね。

◆新番組はパッとしないものがほとんどだが，「料理の鉄人」はけっこういける。「カノッサの屈辱」や「TVブックメーカー」のような，いかにもフジの深夜番組っていうノリが感じられていい。みんなも見てみよう！　Oh!Xの読者ならきっとハマる。　松永　正弘(23)京都府

今度，見てみましょう。私のお勧めは「音楽の正体」フジの水曜深夜ですが，(哲)氏は「音効さん」フジの月曜深夜だそうです。

◆10月21日放送の「平成犬物語バウ」（テレビ朝日PM7：00）でOh!Xの裏表紙にあるX680x0の広告のそっくりさんが登場しました。よく見るとX78000と書いてありました。アニメータのなかにX68000ユーザーがいる可能性が高いので，今後は要チェックです。また，10月26日放送の「トランタン白書」（テレビ東京PM10：00）でも，X68000XVIが登場しました。　武川　昭博(26)神奈川県

それにしてもよく見つけましたね。

◆Oh!X（MZ）とは長いつき合いですが，ハガキを出したのはたぶん初めて。悪いことが起こらなければよいが……。　岩熊　展史(23)福岡県

さあ，あなたにとってこれが載ったのは良いことか悪いことか？

◆CDを凍らすと……CDを布でくるみ，冷凍室に入れて凍らす。次に冷蔵室に移し，一昼夜かけて解凍したのち，おもむろにデッキに入れて聴く……いい音になるらしい。らしいので試していない（笑）。　伊藤　浩克(22)香川県

誰か試してみませんか？　試した方はぜひ報告してくださいね。もちろん凍らせたことで問題が起きても当方としては責任持てませんが……。

◆10月17日，第2種情報処理試験を受験してきました。時間になっても教室の半分が埋まってないのに驚いていたら30分後には3分の1。午後の試験には10人前後しか残っていませんでした。実力よりもとりあえず最後まで残っていれば合格するかもと思い，最後まで残っていましたが結果はいかに？　大畑　佳史(20)福井県

結果はどうだったでしょうか？

◆天気を見ていなかった私は，母の「雨が降る」というひとことを信じてカサを持って学校に行った。だが，いざ駅に行ってみるとだ〜れもカサを持っている人はいない。そのうえ天気もよくなってきた。むなしくなった私は友人に「誰もカサ持ってないねー」と問いかけた。友人は私をかわいそうに思ったのか周りをキョロキョロ見わたして「うしろにいるよ」といった。どんな奴だろうと思って振り向いたところ，今朝一緒に家を出た妹がそこにいた……。　工藤　憲和(17)山形県

その昼には，お母さんがひとり街をカサを持って歩いていたり……はしないですよね。

◆「ブレードランナー（最終版）」をWOWOWでやっていたので見ました。特によかったのはエンディングで流れていた音楽です。なんともいえないミステリアスな音の流れはいままでに聴いた音楽のなかではピカイチです。誰か内蔵FM音源でプログラムしてくれませんかねえ。　中本　孝(20)埼玉県

ビデオでも借りて聴いてみようかな。

◆夫がX68000を中古で買ってくれることになり，再びOh!Xを読み始めました（私の愛機はX1 turboZIII)。夫はMacintoshユーザーなので，ひたすらWindows連合に敵対しているみたいです，私たち。　三木　陽出(33)北海道

いっぱい使ってあげてくださいね。

◆11月号の高橋さんへ。東海道本線でOh!Xを読んでいた50代の男性って私でしょうか。でも私は47歳で，背広も着てません。これから私を見つけたら気楽に声をかけてください。ときどき小田急も利用しますが，電車のなかでは必ずOh!Xを読んでいます。当月号を読みきると古いものを読み返しています。若い方も声をかけてください。X68000orX68030ファン皆さんでX68000を育てましょう。　荒木　昇三(47)神奈川県

これで輪が広がるかもしれませんね。楽しみ楽しみ。

◆本屋に立ち寄るとOh!Xがあった。その本屋には毎月1冊しか入荷していないようなので友達にとっておいてあげようとしばらく待ってた。すると，40歳くらいのおじさんが買っていってしまった。やはりOh!Xは若い読者だけのものではないんだ……。　種井　裕光(17)愛知県

友達はOh!Xを買えたのでしょうか？　それと買われていくOh!Xを見ていた君は買えましたか？　あっ，アンケートハガキがここにあるってことは……。

◆仙台には仙台市科学館というところがあります。そこにはX68000XVIが置いてあって，ジョイスティックを使って館内の案内を見たり聞いたりすることができます。しかもSX-WINDOWのうえで動いている。ところが不思議なことに耳つきのディスプレイを使っていながら別のモノラルスピーカーを前面のPHONEにつないでいて，耳からは音が出ていなかった。　吉田　征二(20)宮城県

この謎の解ける名探偵はどなたかいらっしゃいませんか？

◆プレゼントが当たらないという人が多いようですが，私は応募したうちの1,2割くらいは当たっています。コツとしてはあまり目立たないものを希望する（要するに欲張らない）ことでしょう。やっぱり世の中そんなに甘くはないですね……。　米原　孝太(23)東京都

どうしてもプレゼントを当てたいと思う方は参考にされてはいかが。

◆西武が勝てなかった。翌日，歯医者に行く。高校生にまちがえられた。そのまた翌日，右足がはれて病院へ行く。やはり高校生にまちがえられた。怒る気もしない。ううっ。　岩瀬　貴代美(21)福岡県

社会人だというのに夜の街を歩いていると小学生にまちがえられて補導されてしまう人もいるから，大丈夫ですよ（そうかなあ？）。そういえば編集部でも……。

◆もうすぐ文化祭だ！　そしてあさっては文化祭用のCGのマスターアップの日だ。どうしよう。　川江　武志(22)香川県

締め切りはちゃんと守りましょうね。

◆11月号の「X-OVER・NIGHT」の絵が逆になっていた。ねらいか？　森本　真(18)東京都

そういうことにしておきましょう。

◆自分の努力不足により，留年が決定しました。来年1年間は暇なのでなにかプログラムでも投稿したいと思います。ううっ（涙）。　竹原　充(20)栃木県

◀芹澤　あや子　東京都
「ストIIダッシュ」の突然の移植に驚かれたみたいですが，もうプレイされましたか？　今月号が発売の頃には「餓狼伝説2」の番。楽しみですね。

◀沢地　文彦　兵庫県
続いてこちらのテーマも「ストIIダッシュ」。同じテーマでも描く人によってイラストはこんなに違うんですね。ところで，リー・ラプラダって誰？

来年はがんばってくださいね。

◆友人MはFM放送を聴いていて，次の番組が始まらないのでラジカセの早送りのボタンを押していた。気持ちはわかる。Mよ。

外崎　英基(21)北海道

では，聴きのがしたときは巻戻しを……。

◆めでたく18歳になりました(10月20日)。もう最高のプレゼントがありました。うちの学校では毎年文化講演会があるのですが，1993年は「お笑いマンガ道場」の富永一朗先生だったのです。その日が10月20日で，おまけに先生がその場で描いてくれたマンガをサイン日付，「おめでとう圭一君」の言葉が入ったかたちでいただきました。1993年は最高の誕生日でした。

路川　圭一(18)茨城県

来年もそれに勝る誕生日だといいですね。

◆郵便料金がもうすぐ上がるという。いままで貯めていた小説のアンケートハガキを出そうと思い調べてみた。なんとその数約70枚！　切手代で3,000円以上！　出すのやめようかな……。

伊南　尚幸(18)青森県

これこそ，チリも積もればですね。

◆浪人なのに「メガロマニア」にハマってしまった。「シムシティ」とは違った快感が……。妹の「二郎君と呼ぶぞー」の一言が怖い。

大畑　孝(19)千葉県

2つも名前があるのもいいじゃないですか？　いやなら，いまだったらまだまにあうと思うけど……。

◆しばらくこのハガキを出していなかったのですが「コットン」の湯のみにつられてしまいました。こーゆー人って多くないですか？

百田　浩士(23)大阪府

とっても多いみたいです。11月号のプレゼントでもどうやら一番人気みたいです。

◆近々，手品師としてではないのですが，芸人としてデビューしそうです。

北川　亮(23)東京都

デビューしたら教えてくださいね。応援します。

◆東京電力の電子ちゃんは人妻なのか……。

山口　剛(20)東京都

▲江副　滋　埼玉県
とっても心が落ち着きます。眠っている(?)女の子が安心しきっているみたい。描いていたときは，やさしい気持ちになっていたんでしょうね。

知りませんでした。

◆学祭も無事に終わりました。他校もいろいろとまわろうと思います。目良　美幸(20)大阪府

うらやましい。遊びに行く時間が欲しい。

◆楽あれば楽あり。　長谷部　一也(18)東京都

そのとおり。でもやっぱり苦はあります。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★アイ・オー・データ機器製1Mバイト増設RAMボード「PIO-6BE1-A」（X68000ACE/PRO内蔵用）を6,000円（送料込み）で売ります。箱，説明書など，すべてあります。完動品です。連絡は往復ハガキでお願いします。〒189　東京都東村山市廻田町2-22-8　青野　研(23)

★計測技研製X68000用2Mバイト増設RAMボード「KGB-X68PRK-02」を11,000円で売ります。連絡は往復ハガキでお願いします。〒320　栃木県宇都宮市上大曽町360-1　藤井　雅夫(34)

★システムサコム製X68000用MIDIボード「SX-68M」＋ローランド製サウンドモジュール「MT-32」を30,000円で売ります。共に箱，説明書あり。シャープ製熱転写カラープリンタ「CZ-8PC2」を10,000円で売ります。箱はありませんが，説明書はあります。連絡は往復ハガキでお願いします。〒562　大阪府箕面市半町1-9-6桜井池永寮　越智　亮(21)

★シャープ製熱転写カラープリンタ「CZ-8PC4-BK」，カラーイメージユニット「CZ-6VT1」をそれぞれ25,000円で売ります。連絡は往復ハガキでお願いします。〒020-01　岩手県盛岡市中屋敷町1-65-306　三浦　正宏(23)

★シャープ製カラー漢字プリンタ「CZ-8PG2」(136桁）を65,000円くらいで売ります。箱，ケーブル，マニュアルなど，すべてあります。連絡は往復ハガキでお願いします。〒433　静岡県浜松市上島6-14-8　滝下　真玄(21)

★シャープ製漢字プリンタ「CZ-8PK8」を送料込みで30,000円くらいで売ります。高く買ってくださる方優先です。連絡は往復ハガキでお願いします。〒503　岐阜県大垣市本今3-121　小坂毅俊(29)

★X1用データレコーダ「CZ-8RL1」を付属品，説明書つきで3,000円，シャープ製熱転写カラー漢字プリンタ「MZ-1P17」（黒）を付属品，X1用接続ケーブルつきで6,000円。連絡は往復ハガキでお願いします。〒670　兵庫県姫路市西庄402-2　高島　伸幸(24)

買います

★シャープ製X68000XVI用2Mバイト増設RAMボード「CZ-6BE2A」を送料込み25,000円で買います。説明書はつけてください。連絡は往復ハガキでお願いします。〒799-26　愛媛県松山市福角町625-8　加藤　和人(17)

★シャープ製X68000用1Mバイト増設RAMボード「CZ-6BE1」を送料込み10,000～15,000円で買います。完動品で，箱，説明書，付属品があるもの。連絡は往復ハガキでお願いします。〒355-01　埼玉県比企郡吉見町銀谷81-2　井上　政広(24)

★X68000XVI用2Mバイト増設RAM「CZ-6BE2B」を22,000円で，2個なら50,000円で買います。完動品なら問題ありません。付属品の有無によりプラスアルファします。連絡は往復ハガキでお願いします。〒343　埼玉県越谷市瓦曽根3-6-55　野倉　徹(21)

★MIA出版刊「X1リファレンスノウト」（定価2,500円）を売ってください。往復ハガキに希望価格（定価以上でも可）を書いて送ってください。返事を書きます。往復ハガキ代もお返しします。〒223　神奈川県横浜市港北区日吉本町1-32-7　フルヤハイツ103　宮坂　大也(20)

★MIA出版刊「X1リファレンスノウト」（定価2,500円）を，安価で譲ってください。切り抜き，落書きが少ない本を望みます。連絡は往復ハガキか官製ハガキにてお願いします。〒562　大阪府箕面市桜井2-9-22 宮野文化北201号　河合　匠(23)

バックナンバー

★MZ-2000のS-OS「SWORD」が掲載されている月のOh!X（1987年3月号など）を1,000円で，WICSとBASEの本を3,000～5,000円でお願いします。送料はこちらで負担します。連絡は往復ハガキでお願いします。〒762　香川県坂出市久米町2-6-35　久保　正文(24)

★Oh!X1990年7月号を送料込み1,000円で買います。切り抜き不可。連絡は往復ハガキでお願いします。〒253　神奈川県茅ケ崎市出口町9-3　原　明(44)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は11月号の内容に関するレポートです。

●どこから見ても敷居の高い（ように見える？）システムの場合，とりあえず一度さわらせないといけないわけですが，その意味では，「とりあえず三角錐……」以外はあまり導入役になっていない気がします。まだいまは“〜の活用”でなく“〜の基礎の基礎”くらいで充分すぎるのではないでしょうか。

３Ｄに強く興味を持つパワーユーザーだけ相手にしているのなら別にかまわないのですが……。

石田　伯仁(20) X68030，MZ-731，PC-8801mkⅡMR，PC-E200　神奈川県

●特集を読んで私の数学の知識が足りないと感じた。基本的なことはわかっているはずなのだが，実際，応用編に挑戦！　といったレベルには達していないようだ。おかげで，ポリゴンで「爆発」なんて頭で考えてみても，全然イメージがわかない。

そこで，ある筋では「受験雑誌」として有名な（？）Oh!Xにて，「３Ｄでエクスタシーを得るための数学講座（仮名）」を開いてみてはどうだろうか。同じようなことを望んでいる読者も少なくないと思うのだが……。

吉岡　洋明(20)　X68000 PROⅡ，PC-8801MA，FM-NEW7　埼玉県

●「大人のためのX68000」，お疲れさまでした。この連載は，X68000に対しわりと冷静でPC-9801やMacintosh，IBM PCのことにも触れていて，X68000と距離をおいて読める，好きな連載でしたのに残念です。

最後に荻窪氏が指摘されていますが，「X68000の立場が明確化した」というのは私も同感です。ビジネスに使うのならば，PC-9801のほうが速くて使いやすいし，ゲームをするのならメガドライブやスーパーファミコンのほうが安くすみます。また，CGをやるならばAMIGAのほうがソフトはたくさんあります。現在のパソコンのなかで，自分たちでプログラミングをする楽しさはX68000がいちばん味わえると思います。ところが，Z80のマシンで味わった楽しさまではいかないような気がします。これからのX68000はどうあるべきなのでしょうか？

松永　孝治(23)　XⅠturbo model30，PC-9801N，AMIGA1200/85MB　鳥取県

●以前から欲しいと思っていた「MATIER」に３Ｄステレオグラムの作成ツールがつきました。となれば，すぐにでも買いに行きたいのですが，昨今の経済状態ではそれもままならず，指をくわえてまずはじっと記事を読む……。

ステレオグラム（なぜあんなに流行るのか？）の本は，巷にあふれんばかりですが，それを作成するツール（ソフト）にはあまりお目にかかれません。難しい理論はおいといてお手軽にステレオグラムが作れるとなれば，うれしいし，やってみたいと思います。おまけについてくるところがニクイですね。

野原　賢次(32)　X68000 ACE-HD，XⅠturbo model30　埼玉県

●「Easydraw SX-68K」のレビューは，大変興味深かったです。以前から仕事半分，遊び半分でDTPというものをやってみたいと思っていました。ドローツールの発表により，SX-WINDOWでDTPという夢が見えてきましたが，まだまだ印刷関係が弱そうですね。いまはMacintoshに(DTPだけ)いこうか，SX-WINDOWで待つか悩んでいます。SX-WINDOWでできる環境が揃えば，投資金額も少なくて済むし，X68000シリーズを愛する者としてうれしいかぎりなのですが……。

橋本　和典(26)　X68000XVI　東京都

●「Photo CD」は将来的に見て，すごく楽しみだと思う。いままでも，たしか富士写真フイルムからフロッピーのなかに収めておくのがあった。TVで見られるというやつだが，あれから比べるとすごく実用的で頼もしい製品規格だと思う。５つのフォーマットで入っているのはなにか無駄な感じもするが，それだけ画像もよくなることだし，文句はいうまい。これからのX68000で発展していける分野だと思う。

原田　謙(19)　X68000PRO　石川県

●いままであまり興味のなかった「THE SENTINEL」ですが，今月はマシン語講座というので記事本編のほうまで読んでみました。というのは，「SLASH」がアセンブラでこそ真価を発揮できるというので，アセンブラのプログラミングを構想していました。そこで，これ幸いと飛びついたわけです。16ビットのMC68000よりも８ビットのZ80のほうがアセンブラを身につけやすいだろうと高を括っているのも本当のことですが……。

伊藤氏の記事を読むうちに「これは本当に１からやってくれそうだ」と期待も高まってきました。そしてなによりもこの連載の楽しみなところは，プログラムの楽しさを教わることができそうな点です。他力本願気味になってしまいましたが，これは私がOh!Xに求めていたものです。喉まで出かかっているものが出てきたような気分です。がんばってください。これでパソコンを中身まで楽しめるユーザーが増えるでしょう。

林　大助(18)　X68000SUPER，PC-8801mkⅡFR　神奈川県

●11月号は「CASSAVR.X/CASLOAD.X」がいいですね。実用性はともかく，目のつけどころがシャープですね。こういうふうに，以前からあるものの思いもよらない使い方を見ると楽しくなってきます。こういうプログラムこそアマチュアらしいプログラムだと思います。記事を見ると作者は実用で作られたらしいのですが，実用としてまじめにこういうプログラムを作るというのは，発想が豊かですね。

北風　保(22)　X68000 ACE　東京都

ごめんなさいのコーナー

11月号　ペンギン情報コーナー

P.143　新製品情報として紹介した，パーソナルフィルムレコーダ写譲「FR-1300」を発売している日本アビオニクスの電話番号が間違っていました。正しい番号は，03(3725)7814です。ご迷惑をおかけしましたことをお詫びします。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

どうも お待たせ! ついに発売

▶どうもお待たせしました。ついにというか，やっと「Z-MUSICシステムver.2.0」が発売されました（出ていないとシャレにならない）。最近，編集部で電話をとると3本に1本くらいが「Z-MUSICの新しいのは，もう発売されたんでしょうか」という状況でした。うーん，これで少しは電話から解放されます。

ということで今月の特集は，その発売に合わせてどどーんとまるごと「Z-MUSICシステムver.2.0」でした。

いままで「LIVE in」などの入力方法がわからなかった人も，この記事を読めば音を鳴らせるようになりますね。いままでのリストを打ち込んでみてはいかがですか？ もちろん，そんなことは前から知っているという人にも，バージョンアップされて変わった点や新しいツールなどについて解説してあります。

より詳しくZ-MUSICについて知りたい人は，ディスクとマニュアルの充実した「Z-MUSICシステムver.2.0」を最寄りの書店でよろしく。では，新しいZ-MUSICを使って楽しい曲を作ってください。みなさんが，いい曲を作って送ってくれるのを楽しみに待ってます。

▶今月の新連載は1本です。「"実戦！"ゲーム作りのKNOW HOW」が始まりました。X68000でマシン語のプログラミングを目指す人は，参考になるでしょう。また，当初予定されていた荻窪圭氏の新連載「石の言葉，言葉の夢」は，2月号からスタートに決定しました。

▶さて，2月号といえば毎年恒例になっている，GAME OF THE YEARのノミネート発表が行われます。1993年に発売されたゲームはもちろん覚えていますよね。そこからベストゲームを選びます。アンケートハガキに思い入れのあるゲームへのコメントを書いて送ってください。

▶そして3月号では，年賀状，クリスマスカードのカラーイラストを紹介する予定です。みなさんの気合いの入ったイラストをお待ちしています。もちろんCGでもOKですよ。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「㊢㊀㋮㊔」係

SHIFT・BREAK

▶ついに今年も例のイベントが迫ってきた。さあこい，野郎どもっ，みんなまとめて晴海に大集合だ！俺は待ってるぜ兄貴ーということで，またこりずに同人誌なんぞを作って店を出しております。上昇気流5，132ページ（！）のボリュームで読みごたえは保証付きです。30日（木）ウ-60b「輝竜同盟」をよろしく。絶対来てねー！　(哲)

▶迷惑を省みず，常駐を覚悟でHDDを持ってきたのはいいけれど，マシン室の朝は寒かった……。もっともっと入れたいPCMがあるんだけど，すでにディスクが6枚じゃしょうがないよね。というわけでZの話。で，初めてOutRunnersの曲をまともな音で聴いた。ゲーセンの印象そのまま。やっぱ趣味じゃないな。（スーパーストⅡは非常に理不尽やと思う進)

▶XVIを改造した。といっても，たいした改造ではない。ジョイスティックポートの2番から出ている5ボルトの電極をリレーにつないで，本体背面の電源をPOWERスイッチと連動するようにしただけ。外付けの100MHDとXVIが同時に起動するようになった。Centris610のせいで机に居場所がなくなったXVIも，これで少しは浮かばれる。　(Ats)

▶最近のゲームセンターは空耳のルツボと化している。「餓狼伝説SPECIAL」の不知火舞の必殺技はなんといっているのだろうか？ どうやら「カチョーサン」という技と「ユーメイジン」という技があるようだが。「カチョーサン，ユーメイジン，ニッポンイチ！」これじゃ外国からきたホステスだよ（笑）。ほかの空耳フレーズを知ってる人は教えて。（E.K.）

▶ようやく「悪魔城ドラキュラ」をクリアした。秘訣は「慣れ」といわれているが，特に私のようなヘボプレイヤーの場合は「信じること」が必要と感じた。自分にも必ずできると信じ通すこと。これができぬ者には「悪魔城ドラキュラ」は偉大なクソゲーでしかない（私もクリアするまでそう思っていた）。結論。「悪魔城ドラキュラ」は宗教だ！　(A.T.)

▶最近，自分の中にある世界史感覚を見直す作業を続けている。いかに学校が西欧中心的歴史観を唱えてきたか。やっとそこから脱却しようというのだとすると，キリスト教世界がいかにムチャクチャやってきたかがわかる。十字軍のムチャクチャ。アメリカ大陸へのムチャクチャ。魔女狩りのムチャクチャ。まあ，客観的な歴史なんてものはないのだが。(K)

▶先日，ひどい風邪の症状になり内科に行ったが，いくつかの検査の結果は異状なし。それでも体調が悪いと訴えたが異常がない者は治療できないと却下された。で，精神的なものかと思い，神経科に行くと不安神経症との診断。まさかストレスに悩む現代人の象徴ともいう神経症とは。ちょっと恥ずかしい。みなさんも働きすぎには注意しましょう。　(KO)

▶11月25日，早朝，寒い。仕事が終わって家に戻らず府中へと行く。眩しい朝日が目に染みるなか，移りゆく季節を感じさせる芝の上を異国の馬が駆け抜けていった。ふと，辺りを見ていると，どこかで見たような人が……。あっ，高橋源一郎さんだ。「さっ，サインください」思わずミーハーしてしまった。（レガシーワールドの単勝を持って叫んでいた高)

▶ひさしぶりに観たオペラ「炎の天使」はいわゆるキワモノだった。日本初演だが，いままで上演されなかったのは知名度の低さや作品の評価が分かれるせいではないらしい。上演2日めには警察の立ち会いがあったという。問題は修道女たちが全裸で狂い踊るシーンだが，たったそれだけを邪なキモチで見るのに3万いくらも払う奴なんていないぞ。　(ふ)

▶しばらく前にブームとなった，いわゆる「おいしい水」というやつを初めて買った。いままで，「わざわざ金を出して買うものか」と思っていたのだが，実際に飲んでみたら，それなりにうまく，クセになりそうにまでなっている。確かに砂糖ベタベタのジュースを飲むよりは，格段に体にいいだろう。しかし，なんだか悔しい今日この頃。　(J)

▶「Z-MUSICはいつ出るの？」「SX開発キットが出る頃には……」というのは結構便利ないい回しだったのだが，おかげさまでようやく発売のはこびとなった。すでに店頭に並んでいるはず。諸々の事情により，今回はMookではないので書店で注文するときには注意してほしい。次は支援ツール集とデータ集かな？ 付録ディスクは3月号。　(U)

▶X68000の開発部隊がシャープ幕張研究所に移った。いつかも書いたけど，あのポップアップハンドルのついたビルだ。幕張といえばマルチメディアのメッカ。ただ不況のせいか，今年のマルチメディア'93はちょっとお寒い展示内容だった。アメリカではすでにマルチメディアが商業ベースに乗りつつあるそうだが，どうもピンとこないよね……。　(T)

microOdyssey

走行中に異音を発するようになったので長年使ってきたβデッキに見切りをつけ，８ミリビデオに乗り換えることにした。Ｓ社（無論シャープではない）に入社した知り合いから「こいつぁ気合入ってますよ」と太鼓判を押された機種だ。

画質は期待したほどではなかったが（ノーマルVHSの標準よりやや落ちる程度），テープの小ささには満足している。VHSのテープと並べて「どちらも2時間」といわれれば誰でも感心もするだろう。

何年間かほとんどビデオに触っていなかったためか，あちこちいじりながら録再を繰り返しているだけでも結構遊べる。

また，長いことデジタルな世界に染まっていたためか，「テープのグレードで画質が違う」というのが妙に新鮮に思えたりもした。磁気記憶の情報量が……と考えると，デジタル感覚ではどうもしっくりこない。

YC分離というのは映像回路の基本だが，パソコンをやってる人になら，古典的「くし型フィルタ」などより最近の「3次元デジタルコムフィルタ」のほうが説明しやすい。昔は職人さんがガラス遅延管で組んでいたなどというと目を丸くされてしまう。

最近はビデオ機器でもデジタル処理用にCCDなどを使っていることも多いが，これが１画素あたり８～10ビットに相当する。これがもちろん無圧縮のフルカラーなのだからアナログの世界というのは，実は凄まじいことをやっているものだ。いや，デジタルのほうが無駄だらけなのか？

「この互換機，CD並みのサンプリング音源積んでるのにどうしてこんなに音が汚いの？」「あっちの人は気にしないんじゃないですか」「いいえ，よく見てください。傍に凄いノイズ源があるでしょ」

ああ，確かに。このあたりはちゃんとアナログの設計をやってないとどうしようもない部分だ。

以前，どうしてデジタル記録のCDが機種によって音が違うのか話題になったことがある。これは「○○社のFDに入れるとプログラムが速くなる」というのと同じくらいおかしなことのはずなのだ。

ビデオでは使う人の気合が画質に出てくる。F社のHGテープとHiFiテープは同じ工程で作られる（最近は知らないが）。一定のチェックを行ったのち出荷されるのがHiFiでノーチェックなのがHGとなる。物理特性は基本的に同じである。どちらかといえば，気持ちの問題だ。

エアチェックなどで気合を入れるときはHiFiを使う。不思議に画質が違う。高価なテープだからと，普段以上のメンテナンスを行うことが影響しているのだろう。筐体の振動防止，部屋中の電気製品を消し，電源ノイズを抑え，１時間くらいチューナー調整を行う。気合の反映するアナログの感覚は楽しい。

件の知り合い曰く，「あのビデオ，マイナーチェンジすることに決まりました。操作系がまるっきり変わります」おい，出荷されてまだ半月しかたってないぞ。買う前に最近のＳ社の初期ロットは大丈夫かとちゃんと確認したじゃないか……。 （U）

1994年２月号１月18日（火）発売

特集　X-BASICとグラフィック

・X68000のグラフィック機能を使う

1993年度　GAME OF THE YEARノミネート発表

荻窪圭の新連載「石の言葉，言葉の夢」

THE SOFTOUCH

ストリートファイターⅡダッシュ/餓狼伝説２/キーパー

全機種共通システム

S-OSで学ぶZ80マシン語講座（３）

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートBI	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ５	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101　東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X　１月号

■1994年１月１日発行　定価600円（本体583円）
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103　東京都中央区日本橋浜町3-42-3
Oh!X編集部　☎03(5642)8122
販売局　☎03(5642)8100　FAX 03(5641)3424
広告局　☎03(5642)8111
■印　刷　凸版印刷株式会社

バックナンバー案内

BACK ISSUES

ここには1993年1月号から12月号までをご紹介しました。現在1992年6，7，9，12，1993年6～12月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は156ページを参照してください。

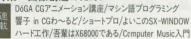
1993

1月号（品切れ）
特集　D.I.Y.ハードウェア

連載
DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
大人のためのX68000/ハード工作/Computer Music入門

●新製品紹介　サンダーワード/SX広辞苑
LIVE in '93　ムーンライト伝説/チャコの海岸物語
THE SOFTOUCH　オーバーテイク/ストライダー飛竜/エアーマネジメント/パイプドリーム 他
全機種共通システム　実践Small-C講座(9)EDC-Tの拡張

2月号（品切れ）
特集　画像創造のために

連載
DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　Communication SX-68K
LIVE in '93　FIRE CRACKER/サンバDEグワッシャ!
THE SOFTOUCH　極/ドラゴンスレイヤー英雄伝説/機甲装神ヴァルカイザー/キングス・ダンジョン
全機種共通システム　BLACK JACK

3月号（品切れ）
特集　X-BASICを学ぶ

連載
DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ANOTHER CG WORLD/ハード工作
ショートプロ/Computer Music入門/Z80's Bar

●緊急速報　32ビットマシンX68030
●新製品紹介　音源モジュールSC-33/GS音源搭載JW-50
LIVE in '93　ストリートファイターII/晴れたらいいね 他
THE SOFTOUCH　究極タイガー/チェルノブ/シムアント 他
全機種共通システム　シューティングゲームコアシステム作成法(1)

4月号（品切れ）
特集　X68第7世代へ

連載
DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
ハード工作/吾輩はX68000である/Computer Music入門

●決定!　1992年GAME OF THE YEAR
●名作ゲーム再遊記
LIVE in '93　FIGHTMAN/ミンキーモモより 愛しのマーシカ
THE SOFTOUCH　スターフォース/元朝秘史 他
全機種共通システム　シューティングゲームコアシステム作成法(2)

5月号（品切れ）
特集　襲撃!　SX-WINDOW
第8回　言わせてくれなくちゃだワ

連載
DōGA CGアニメーション講座/ANOTHER CG WORLD
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

●X68030へのソフトウェア対応について
LIVE in '93　MAGICAL SOUND SHOWER/もう笑うしかない 他
THE SOFTOUCH　エトワールプリンセス/メガロマニア 他
全機種共通システム　シューティングゲームコアシステム作成法(3)

6月号
創刊11周年特別企画　確率遊技シミュレーション

連載
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　SC-55mkII
LIVE in '93　ストリートファイターIIより 春麗のテーマ/BAY YARD/LOVE&CHAIN
THE SOFTOUCH　餓狼伝説/信長の野望・覇王伝 他
全機種共通システム　REVERSI

7月号
特集　席巻するローテク文明

連載
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/マシン語プログラミング
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　ドローイングパット33070&MATIER
LIVE in '93　Midnight Circle/今日の日はさようなら/赤い靴
THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/大航海時代II/銀河英雄伝説III/幻影都市/ヴェルスナーグ戦乱
全機種共通システム　MSX用S-OS "SWORD"

8月号
特集　C言語実践的入門

連載
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/Computer Music入門/大人のためのX68000
吾輩はX68000である/ショートプロ/ANOTHER CG WORLD

●特別企画　夏真っ盛り，アマチュアリズムのX68000
LIVE in '93　SPLASH WAVE
THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/餓狼伝説/ロボットコンストラクションR.C./Winning Post
全機種共通システム　MACINTO-C再掲載

9月号
特集　光学式磁気円盤MO

連載
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/Computer Music入門/ANOTHER CG WORLD

●新製品紹介　OS-9/X68030
LIVE in '93　ファイナルファンタジーVのテーマ/銀河鉄道999/アルスラーン戦記IIより 汗血公路/ちょうちょ
THE SOFTOUCH　悪魔城ドラキュラ/コットン/ダーク・オデッセイ 他
全機種共通システム　7並べ/SLANG再々掲載

10月号
特別企画　秋祭りPRO-68K

連載
ハードコア3D/Computer Music入門/マシン語プログラミング
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/吾輩はX68000である

●特別付録　秋祭りPRO-68K（5″2HD）
●SCSIパックンTOWER JACK
LIVE in '93　未来予想図II/OutRunより PASSING BREEZE
THE SOFTOUCH　コットン/The World of X68000/あにまーじゃんV3
全機種共通システム　シューティングゲームコアシステム作成法(4)

11月号
特集　ポリゴナイザSLASHの活用

連載
ハードコア3D/Computer Music入門/ファイル共有の実験と実践
こちらシステムX探偵事務所/目指せジョイスティックの星
響子 in CGわ～るど/ショートプロ/大人のためのX68000

●新製品紹介　Easydraw SX-68K
OS-9 Ultra C/Technical Tool Kit
LIVE in '93　渚のアデリーヌ/エロティカ・セブン
THE SOFTOUCH　ぶたさん/ダイアット・ヴァークス
全機種共通システム　S-OSで学ぶZ80マシン語講座(1)

12月号
特集　古今東西ゲーム議論

連載
ハードコア3D/マシン語プログラミング/響子 in CGわ～るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

●新製品紹介　MATIER ver.2.0
C Compiler PRO-68K ver.2.1 NEW KIT
LIVE in '93　クリスマス・イブ/星に願いを
THE SOFTOUCH　ネメシス'90改/項劉記/スーパーリアル麻雀PII&PIII
全機種共通システム　エディタアセンブラREDA再掲載

FiLo ファイロ ブラックモデル現る!!

エアフィルタ交換不要の3.5インチ光磁気ディスクユニット

X680x0にジャストフィット

■世界最小クラスのコンパクトなボディ。縦置/横置可能。
■深夜でも気にならない低騒音ファンを使用。
■平均シークタイム30ms、回転数3600rpmの高性能ドライブ。

CS-M120PX　通販限定

本製品は、特別企画商品につき一般の小売店では購入できません。

標準価格 178,000円 → 特別価格 118,000円（税込）

●お申し込みはFAXまたは郵送にて

注文書の太枠線内にご記入の上郵送またはFAXにてお送りください。

お申し込み先
コパル綜合サービス株式会社　通販係
東京都板橋区志村2-16-20
TEL 03-3965-1144
FAX 03-3558-3229

●お支払いは銀行振込で

代金は下記の口座までお振込みください。
（振込手数料はお客様負担で電信扱でお振込ください。）

口座番号　東海銀行　板橋支店　当座預金160141
口座名義　コパル綜合サービス株式会社

・商品の引き渡しは代金お支払い後となります。
・商品はご入金確認後、原則として3日以内に発送致します。
（在庫切れの場合はご連絡いたします。）

◆今回お買い求めの方に限りケーブル*・ターミネータ・送料をサービス。
***ご注文の際にご希望のケーブルをご指定ください。**

◆SCSI I/Fボードはパソコン本体に付属のものまたは純正品が使用可能です。
その他サードパーティ製のSCSI I/Fボードとの接続についてはお問い合わせください。

X680x0以外のパソコン用接続キット、オプションも用意しております。

主な接続キット
- ●PC98接続キット
- ●Macintosh接続キット
- ●FM接続キット
- ●AT接続キット

※商品の技術的なご質問・ご相談は
ユーザーサポート係　TEL03-3965-1161

FiLo注文書　FAX 03-3558-3229

品名	CS-M120PX	ご注文台数　　台	ご連絡先 TEL（　） FAX（　）
ケーブル*1	□フル～ハーフ　□ハーフ～ハーフ		
お名前	フリガナ		
お届先住所	（〒　-　）都道府県　区市郡	1．会社　2．自宅	

（弊社記入欄）
受付番号
受付日
納入日
備考

*1 どちらかご希望のケーブルをご指定ください。

2・2倍速CD-ROMドライブ KGU-XCD2同等品 限定特価¥68,000で好評発売中!!（ドライバ別売り）

for X680x0 Series Only

オリジナル アプリケーション

開発速報#5

R&D Division of 計測技研 FirstClassTechnology

新発売

SX-WINDOW用スケジュール管理ソフト

DoubleBookin' 定価 ¥12,800

SX-WINDOW だから、いっしょに暮らせる

SX-WINDOW。そのマルチタスク環境では、ユーザーの作業環境は一変します。より自由に、感性のおもむくままに。SX-WINDOWとDoubleBookin'の組み合わせで、あなたの暮しが、仕事が、変わります。

DoubleBookin'は、あたらしい考え方にもとづいたスケジュール管理ソフトです。

●多角的な予定設定/表示

予定はカレンダー、デイリー、グラフの3つのウィンドウで確認することができ、多角的に検討することができます。

何日にもまたがる長期の予定にも対応。短期の予定とあわせて、日常生活に即したスケジュール設定が可能です。

●マルチタスクをいかした豊富なイベント

予定を設定した時刻をメッセージやアラームで知らせるのはあたりまえ。DoubleBookin'は、SX-WINDOWのマルチタスク環境をいかして、様々なイベントを起こすことができます。音楽を演奏したり、テレビ画面に切り替えたり、シャーペンやEasy drawを起動したり…など、思うままのライフスタイル設計を可能にします。

●モジュール追加で成長し続けます

予定に設定したイベントの種類は、外部モジュールを追加することによってさらに増やすことができます。

今後新しいアプリケーションや周辺機器が登場した場合でも、DoubleBookin'はそれらを取り込んで成長し続けます。

●電子手帳やシステム手帳とリンク

シャープ製電子手帳(DB-Z以上)とのスケジュールのやりとりを可能にしたほか、カレンダーやスケジュールのリフィル印刷もサポート。自宅やオフィスに縛られることなく、幅広いフィールドで活用できます。

●アンフィニーシステム社製MIC-68Kにも正式対応

ユーザウィンドウ
ディスク容量が許す限り、何人のスケジュールでも登録可能。

カレンダーウィンドウ
スケジュールの設定状況が、ひと目で確認できるカレンダー表示。ここが作業の中心です。

デイリーウィンドウ
カレンダー上の日付をダブルクリックすると、一日のスケジュールを詳しく表示します。

編集ウィンドウ
スケジュールの入力はとてもかんたん。イベントアイコンを選んで予定の内容を記入するだけ。

発売中

CD-ROM Driver Ver1.06 定価 ¥4,800

CD-ROM Driver Ver1.06は、Human68k上でCD-ROMをフロッピー感覚で扱えるようにするデバイスドライバです。一般的なISO9660フォーマットで記録されているCD-ROMなら、X680x0でそのまま読み出すことができます。

(1)ISO9660アクセス対応ドライブ
市販されているほとんどのCD-ROMドライブ

(2)CD-ROM XA対応ドライブ(SX-PhotoGallery対応)

東芝	XM-3301、XM-3401、およびその互換製品
ロジテック	LCD-500
松下電器	LK-RC533N25
弊社	KGU-XCD、KGU-XCD2

(3)オーディオ対応ドライブ

東芝	XM-3301、XM-3401、およびその互換製品
各社	SCSI-2コマンド対応製品
弊社	KGU-XCD、KGU-XCD2

発売中

SX-WINDOW用Photo-CDビュアー

SX-PhotoGallery 基本セット¥15,000

大切な写真をいつまでも美しく保存できるPhotoCD。SX-PhotoGalleryなら、PhotoCDのフルカラー記録をSX-WINDOW Ver.3.0のグラフィックウィンドウ上に再現できます。

SX-WINDOWの特長である、カット&ペーストによるアプリケーション間でのデータのやりとりにも対応。また,PhotoCDの画像展開モジュールはIVM.X用のリソースとして用意しましたので、キャンバス、シャーペン、Easydraw、EasypaintなどでPhotoCD画像を利用することができます。様々なソフトと連係し、力をあわせ、その結果のクオリティを追求するSX-WINDOWの思想を踏襲しています。

SX-PhotoGalleryの機能をすぐにお試しいただけるよう、Photo-CDのサンプル画像を収めた「Kodakフォトサンプラー」バンドルセット(定価¥19,000)もご用意いたしました。

※ SX-PhotoGalleryにはCD-ROM Driverが付属します。

発売中

X680x0用フリーソフトウェア集CD-ROM

FreeSoftwareSelection Vol.1 定価 ¥5,000

お求めはお近くのパソコンショップ、または弊社通販部(TEL:0286-22-9811)へお申し込みください。

低金利クレジット　通信販売送料 全国一律¥1,000　長期クレジット可能

※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE 本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

号外！ FreeSoftwareSelection Vol.2プロジェクト始動！ フリーソフト作家の皆さん、準備はいいですか？ (^O^)／オーッ

マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス!

SHARP

注目!! 平成6年3月末一括払い手数料(金利)無料(平成6年1月末/2月末/3月末のいずれかをご指定ください。)

12/18〜1/17

X68000 Compact XVI

旧シリーズ今が買いどき!!
(クレジット表:送料・消費税込み)送料¥2,000・消費税別

①本体+モニター

- CZ-674C-H
- CZ-608D-H

定価¥392,800

P&A超特価 ¥162,000

12回	14,800	24回	7,800	36回	5,400	48回	4,300	60回	3,600

②本体+モニター+FDD(5"×2)

- CZ-674C-H
- CZ-608D-H
- CZ-6FD5(FDD)

定価¥492,600

P&A超特価 ¥209,000

12回	19,100	24回	10,100	36回	7,000	48回	5,500	60回	4,600

③本体+モニター(TVチューナー付)

- CZ-674C-H
- CZ-614D-TN
- CZ-6CR1(RGBケーブル)
- CZ-6CT1(TVコントロール)

定価¥443,000

P&A超特価 ¥199,000

12回	18,200	24回	9,600	36回	6,700	48回	5,200	60回	4,400

④本体+モニター(TVチューナー付)+FDD(5"×2)

- CZ-674C-H
- CZ-614D-TN
- CZ-6CR1(RGBケーブル)
- CZ-6CT1(TVコントロール)
- CZ-6FD5(FDD)

定価¥542,800

P&A超特価 ¥247,000

12回	22,500	24回	11,900	36回	8,300	48回	6,500	60回	5,400

X68000 XVI

①本体+モニター

- CZ-634C-TH(本体)
- CZ-608D-H(モニター)

定価¥462,800

P&A超特価 ¥213,000

12回	19,500	24回	10,300	36回	7,100	48回	5,600	60回	4,700

モニター変更の場合

※Compact XVI ①・②/XVI ①のモニターを、
- CZ-607D-TN(定価¥99,800)に変更の場合¥3,000加算して下さい。
- CZ-621D(B)(定価¥168,000)に変更の場合¥58,000加算して下さい。

X68030/68000メモリボード(I/Oデータ)

①SH-5BE4-8M(X68030用)……………(送料・消費税込み¥47,586)特価¥45,500
②SH-6BE1-1ME(600C専用)……………(送料・消費税込み¥12,669)特価¥11,600
③1MB増設RAMボード(ACE/PRO/PROII用)(送料・消費税込み¥12,669)特価¥11,600
④2MB増設RAMボード(拡張スロット用)・(送料・消費税込み¥24,411)特価¥23,000
⑤4MB増設RAMボード(拡張スロット用)・(送料・消費税込み¥40,170)特価¥38,300

モデム

(送料¥1,000)

マイクロコア ●MC-14400FX……………(定価¥46,800)▶特価¥34,500
富士通 ●FMMD-3111G……………(定価¥35,800)▶特価¥24,800
オムロン ●MD-24XT10V……………(定価¥29,800)▶特価¥22,500
●MD-96XT10V……………(定価¥46,800)▶特価¥32,000
アイワ ●PV-AF144V5……………(定価¥64,800)▶特価¥49,000

●本広告の掲載の商品の価格については、消費税は含まれておりません。

X68030お買い得セット

(クレジット表:送料・消費税込み)

①X68030

●CZ-500C ●CZ-608D

定価合計¥492,800

P&A超特価 ¥329,000

12回	30,000	24回	15,900	36回	11,000	48回	8,600	60回	7,200

②X68030 HD

●CZ-510C ●CZ-608D

定価合計¥582,800

P&A超特価 ¥409,000

12回	37,300	24回	19,700	36回	13,700	48回	10,700	60回	9,000

③X68030 Compact

●CZ-300C ●CZ-608D

定価合計¥482,800

P&A超特価 ¥340,000

12回	31,000	24回	16,400	36回	11,000	48回	8,900	60回	7,500

④X68030 Compact HD

●CZ-310C ●CZ-608D

定価合計¥572,800

P&A超特価 ¥403,000

12回	36,700	24回	19,400	36回	13,500	48回	10,500	60回	8,900

■モニターの変更

(※300シリーズにチューナー付のモニターを接続の場合CRTケーブルを購入して下さい。)

①CZ-607D(チューナー付)に変更の場合¥ 3,000
②CZ-614D(チューナー付)に変更の場合¥31,000
③CZ-621D(B)…………に変更の場合¥58,000
} 加算して下さい。

X68030 発売記念

X68030をモニターとセットで/単品で } 購入の方

さらに現在お持ちのパソコンと、下取り交換されたお客様に期間中もれなく、
①サイバーステック (CZ-8NJ2 ¥23,800)
②X-68000 フロッピーアタッシュケース (¥8,000)
とクリスタルポルシェ (¥8,000)
以上のいずれかプレゼント!!

①

②

X68000/68030専用ハードディスク

(送料¥1,000・消費税別)

外付

■富士通

⊙FMHD-1201G(120MB、17ms)‥定価¥ 70,000▶特価¥49,800
⊙HD-K200(モッキンバード)(200MB、13ms)・定価¥ 79,800▶特価¥46,800
⊙HD-K240(モッキンバード)(240MB、15ms)・定価¥ 79,800▶特価¥49,800
⊙HD-K540(モッキンバード)(540MB、10ms)・定価¥148,000▶特価¥98,000

■ロジテック

⊙SHD-FMX120(120MB)(ケーブル付)
……………定価¥ 59,800▶特価¥47,000
⊙SHD-FMX240(240MB)(ケーブル付)
……………定価¥138,000▶特価¥57,800

■ジェフ

⊙GF-240e(240MB/15ms/64K)・定価¥118,000▶特価¥ 59,800
⊙GF-340i(340MB/14ms/64K)‥定価¥158,000▶特価¥ 87,800
⊙GF-540i(540MB/8.5ms/256K)定価¥238,000▶特価¥151,800

内蔵

■CZ-500C/300C専用

⊙CZ-5H08(80MB/23ms)
……………定価¥ 98,000▶特価¥71,800
⊙CZ-5H16(160MB/18ms)
……………定価¥135,000▶特価¥99,500

ズバリご奉仕

P&Aならではの 新品パソコン5年保証

《業界No.1の"P&Aメンテナンスサポート"》

最高の保証システム

① 業界最長の新品パソコン5年保証（※モニター・プリンター3年間保証.// ※一部商品は除きます。）
② 中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証.//）
③ 初期不良交換期間3ヶ月（※新品商品に限らせていただきます。）
④ 永久買取保証
⑤ 配達日の指定OK//（土曜・日曜・祭日もOK.//）
⑥ 夜間配達もOK//（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

① 翌月一括払い手数料無料（ご利用下さい。）
② 業界No.1の低金利//
③ 月々の支払いは￥1,000より
④ 9ヶ月先からのスキップ払いOK//
⑤ 84回までの分割、ボーナス併用OK//
⑥ カレッジクレジット
⑦ ステップアップクレジット
⑧ ボーナスだけで10回払いOK//
⑨ 現金一括支払いOK//
⑩ 商品到着払いOK//（代引き手数料が必要になります。10万円まで900円）（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け リースシステム
業務に最適なシステムを構築します。損金処理が可能なリース契約をどうぞ。

周辺機器コーナー （送料￥1,000・消費税別）

カラーイメージスキャナ

■JX-220X《限定》
定価￥168,000
特価￥89,800

■JX-325X
定価￥190,000
特価￥143,000

カラーイメージジェット

■IO-735X-B
定価￥248,000
特価￥128,000

FDD（5インチ×2基）

■CZ-6FD5
定価￥99,800
P&A超特価 ￥49,800

漢字プリンター（ケーブル用紙付）

■CZ-8PC5-BK
定価￥96,800
▶特価￥38,000

■CZ-8PK10
定価￥97,800
▶特価￥71,000

光磁気ディスク（X68000用）

■CS-M120（コパル）
●ケーブル、ターミネータ付 ￥178,000
特価￥119,000

■LMO-FMX330
●ケーブル、ターミネータ付 ￥178,000
特価￥135,000

- ●CZ-8NS1…………定価￥188,000▶特価￥133,000
- ●CZ-6VT1…………定価￥ 69,800▶特価￥ 49,500
- ●CZ-6TU…………定価￥ 33,100▶特価￥ 23,900
- ●BF-68PRO………定価￥ 19,800▶特価￥ 14,400
- ●CZ-8NM3…………定価￥ 9,800▶特価￥ 7,200
- ●CZ-8NT1…………定価￥ 13,800▶特価￥ 10,000
- ●CZ-6BE2A………定価￥ 59,800▶特価￥ 42,800
- ●CZ-6BE2B………定価￥ 54,800▶特価￥ 39,300
- ●CZ-6BE2D………定価￥ 54,800▶特価￥ 39,300
- ●CZ-6BF1…………定価￥ 49,800▶特価￥ 35,800
- ●CZ-6BP1…………定価￥ 79,800▶特価￥ 57,000
- ●CZ-6BM1A………定価￥ 26,800▶特価￥ 19,300
- ●AN-S100…………定価￥ 36,600▶特価￥ 26,300
- ●CZ-6SD1…………定価￥ 44,800▶特価￥ 32,500
- ●CZ-6BN1…………定価￥ 29,800▶特価￥ 21,500
- ●CZ-6BV1…………定価￥ 21,000▶特価￥ 15,200
- ●CZ-6BC1…………定価￥ 79,800▶特価￥ 57,000
- ●CZ-6BG1…………定価￥ 59,800▶特価￥ 43,000
- ●CZ-6BU1…………定価￥ 39,800▶特価￥ 28,500
- ●CZ-6PV1…………定価￥198,000▶特価￥142,000
- ●CZ-6BS1…………定価￥ 29,800▶特価￥ 21,500
- ●CZ-8NJ2…………定価￥ 23,800▶特価￥ 17,500
- ●CZ-6BL2…………定価￥298,000▶特価￥214,000
- ●CZ-6CS1(674C用)定価￥ 12,000▶特価￥ 8,900
- ●CZ-68HA……………………▶特価￥ 91,000
- ●CZ-6CR1（RGBケーブル）…………定価￥ 4,500▶特価￥ 3,600
- ●CZ-6CT1（テレビコントロール）…………定価￥ 5,500▶特価￥ 4,400
- ●CZ-6BP2…………定価￥ 45,800▶特価￥ 33,300
- ●CZ-5MP1（X68030用）…………定価￥ 54,800▶特価￥ 42,000

■システムサコム ボード
- ●SX-68MII（MIDI） 定価￥19,800▶特価￥13,500
- ●SX-68SC（SCSI） 定価￥26,800▶特価￥17,500

（X68030用）
- ●CZ-5BE4 定価￥54,800 ▶￥42,000
- ●CZ-5ME4 定価￥49,800 ▶￥38,000

X68000用ソフトコーナー （送料￥700・消費税別）

- ●Z's STAFF PRO68K Ver.3.0（ツァイト）…………定価￥58,000▶特価￥37,500
- ●Z's TRIPHONYデジタルクラフト（ツァイト）…………定価￥39,800▶特価￥27,000
- ●テラッツォ（ハミングバード）…………定価￥19,400▶特価￥13,600
- ●ラジックパレット（ミュージカルプラン）…………定価￥19,800▶特価￥14,200
- ●たーみのる2（SPS）…………定価￥17,800▶特価￥13,000
- ●Mu-1 Super（サンワード）…………定価￥39,800▶特価￥28,500
- ●CMA68K（シティソフト）…………定価￥29,800▶特価￥21,800
- ●サイクロンEXPRESS α68…………定価￥98,000▶特価￥69,000
- ●C-TRACE68 Ver.3.0（キャスト）…………定価￥98,000▶特価￥68,500
- ●OS-9/X68030 V.2.4.5（マイクロウェアシステムズ）…………定価￥25,000▶特価￥19,900
- ●C&Professional Pack V.3.2（マイクロウェアジャパン）…………定価￥80,000▶特価￥57,800
- ●ウエットペイント1〜3（ウエーブトレイン）（各）…………定価￥15,000▶特価￥11,500
- ●マチエール Ver.2.0…………定価￥39,800▶特価￥28,800
- ●Windex PRO68（JEL）…………定価￥28,000▶特価￥20,500
- ●CZ-213MSD MUSIC PRO68K…………定価￥18,800▶特価￥13,200
- ●CZ-214MSD SOUND PRO68K…………定価￥15,800▶特価￥11,300
- ●CZ-215MSD Sampling PRO68K…………定価￥17,800▶特価￥12,500
- ●CZ-220BSD DATA PRO68K…………定価￥58,000▶特価￥40,000
- ●CZ-225BSV Multiword Ver.2.0…………定価￥32,000▶特価￥23,000
- ●CZ-243BSD CYBERNOTE PRO68K…………定価￥19,800▶特価￥15,000
- ●CZ-247MSD MUSIC PRO68K（MIDI）…………定価￥28,800▶特価￥20,500
- ●CZ-249GSD CANVAS PRO68K…………定価￥29,800▶特価￥22,000
- ●CZ-251BSD Hyperword…………定価￥39,800▶特価￥29,400
- ●CZ-253BSD CARD PRO68K Ver.2.0…………定価￥29,800▶特価￥22,700
- ●CZ-257CSD Communication PRO68K Ver.2.0…………定価￥19,800▶特価￥15,300
- ●CZ-258BSD Teleportion PRO68K…………定価￥22,800▶特価￥16,900
- ●CZ-261MSD MUSICstudio PRO68K Ver.2.0…………定価￥28,800▶特価￥21,200
- ●CZ-263GWD Easypaint SX-68K…………定価￥12,800▶特価￥ 9,800
- ●CZ-264GWD Easydraw SX-68K…………定価￥19,800▶特価￥15,300
- ●CZ-265HSD NewPrint Shop Ver.2.0…………定価￥20,000▶特価￥15,400
- ●CZ-266BSD PressConductor PRO68K…………定価￥28,800▶特価￥22,000
- ●CZ-267BSD CHART PRO68K…………定価￥38,000▶特価￥29,800
- ●CZ-272CWD Communication SX68K…………定価￥19,800▶特価￥14,500
- ●CZ-275MWD SOUND SX68K…………定価￥15,800▶特価￥11,500
- ●CZ-284SSD OS-9/X68000 Ver.2.4…………定価￥35,800▶特価￥25,600
- ●CZ-286BSD BUSINESS PRO68K…………定価￥28,000▶特価￥20,500
- ●CZ-288LWD開発キット（workroom）…………定価￥39,800▶特価￥29,700
- ●CZ-290TWD SX-WINDOW ディスクアクセサリー集…………定価￥14,800▶特価￥11,500
- ●CZ-294SS（5"）/SSC（3.5"） SX-WINDOW Ver.3.0…………定価￥19,800▶特価￥15,200
- ●CZ-295LSD C-Compiler PRO68K Ver.2.1 NEW KIT…………定価￥44,800▶特価￥32,500

☆ゲームソフト25%OFF OK.//（一部ソフト除く）

P&A 株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10：00〜PM7：00 日・祭：AM10：00〜PM6：00
☎03-3651-0148（代）
FAX.03-3651-0141
●定休日／毎週水曜日

全国通販

★頭金なし！
★即日発送

●お近くの方はお立寄り下さい。専門係員が説明いたします。
●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
●ビジネスソフト定価の20%引きOK！TELください。

P&A特選 今月の中古特選品

- ●CZ-600C‥￥55,000
- ●CZ-601C‥￥65,000
- ●CZ-611C‥￥70,000
- ●CZ-652C‥￥75,000
- ●CZ-612C‥￥95,000
- ●CZ-603C‥￥85,000
- ●CZ-653C‥￥78,000
- ●CZ-612C‥￥90,000
- ●CZ-623C‥￥110,000
- ●CZ-674C‥￥108,000
- ●CZ-634C‥￥130,000
- ●CZ-644C‥￥178,000

※上記は単品価格、モニター別売。

新古品 限定
●CZ-674CH
●CZ-608DH
￥138,000

中古品
●CZ-674CH
●68000専用モニター付
￥128,000

限定
●CZ-634CTN（チタン）（中古）
●CZ-613D（グレー）（新品）
￥190,000
（モニターをCZ-614TN（チタン）に変更の場合￥20,000加算）

中古品
●CZ-634CTN
●68000専用モニター付
￥158,000

新古品 限定
●CZ-644CTN
●CZ-604DB
￥228,000

中古品
●CZ-644CTN
●68000専用モニター付
￥198,000

中古・高価現金買取り／下取りOK//

■まずはお電話下さい。
下取り専用
買取り電話 ▶ ☎03-3651-1884 FAX.03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送りください。

買取り価格…完動品・箱／マニュアル／付属品の価格です。

●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
●買取りの場合…現品が着き次第、2日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。
●近郊の方はP&A本店に直接お持ちください。即金にて￥1,000,000までお支払い致します。

●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
●本商品の掲載の商品の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&A特選パソコンラック&OAチェアー （消費税込み）（送料無料、離島を除く）

① 3段 ￥8,240
② 4段 ￥9,785
③ 5段 ￥11,845

※全機種→キャスター付
※上から2番目棚板移動可能（4/5段）
※フレーム色：4段→黒、3/5段→ホワイト

① ￥9,270
●布張り（ダーククレー）
●カスシリンダー

② ￥11,330
●布張り（ダーククレー）
●カスシリンダー
●肘付

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回〜84回払いまで出来ます。但し、1回のお支払い額は¥1,000円以上。

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 （株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.9	3.9	4.9	5.4	8.4	11.4	15.9	20.9	26.9	34.9

南口 徒歩2分
至秋葉原
新小岩駅
JR
バスターミナル
住友BK
東海BK
リブ
北海道拓殖BK
ローソン
P&A新本店
蔵前通り
至千葉

※お支払いは、便利な商品到着払い〈手数料（10万円まで900円）要〉をご利用下さい。

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上お申し込み下さい。詳しくは、お電話でお問い合せ下さい。

POLYPHON

X68000 サブMPUボード 〜ポリフォン〜

■POLYPHONはアクセラレータではありません！

POLYPHONはサブMPUボードです。アクセラレータと異なりメインのMPUには干渉されません。従って、メインとは別のタスクとして処理できます。ですからPOLYPHON用のアプリケーション実行させながら、別のプログラムをX68000本体で実行するといったことも可能となります。ポリフォンシステムとの組み合わせにより、DoGA(REND.X)やGCC・HAS・HLKなどの実行ファイルもX68000本体と同時に別タスクとして動作可能。POLYPHON-24使用時にはパフォーマンスが約2.0〜約2.4倍に向上します。

■POLYPHONはメモリボードにもなります

POLYPHON上にはサブMPUが使用する2MBの他にX68000本体用のメモリを最大8MB搭載できます(0MB/8MBモデルとして販売)。本体用メモリ部分は純正メモリボード同様に使用できます(サブ用メモリはどちらのモデルも2MBですが、こちらは増設できません)。

■POLYPHONはコプロボードにもなります

POLYPHONはコプロを装着することが出来ます(コプロ付モデルは装着済)。コプロ部分は純正互換ですので、FLOAT3などで簡単に利用することが出来ます。

■POLYPHONはMIDIボードにもなります

POLYPHON上にはMIDIコネクタを装備(2IN/1OUT)しています。残念ながらこちらは純正非互換ですが、Z-MUSICをはじめとする各種ミュージックドライバーもPOLYPHONのMIDI OUTをサポートしているので安心です。また、市販ソフトに関してはPOLYPHON-MIDI対応パッチを用意していますので、こちらを利用すれば問題なく利用できます。(パッチはPOLYPHONシステムディスクに付属)(市販ソフトでもZ-MUSIC対応ならば、Z-MUSICの差替えのみで動作します)

■本体にない付加機能も提供します

POLYPHONには本体にない機能としてステレオPCM機能を提供しています。POLYPHON上にステレオ出力端子を備え、高品質にPCMを再生します。

POLYPHON標準価格

POLYPHON	メインメモリ8MBモデル	¥85,000-
POLYPHON	メインメモリ8MBモデル(68881付)	¥95,000-
POLYPHON	メインメモリ0MBモデル	¥62,000-
POLYPHON	メインメモリ0MBモデル(68881付)	¥72,000-

POLYPHON-24の出荷は12月中旬以降のロット分からとなっております。それ以前にお買い求めになられたユーザーの方のために、クロックモジュールアップグレードを用意しております。準備が整い次第、購入ユーザーの方には案内状を送付いたしますので、今しばらくお待ちください。

POLYPHONシステムディスクのバージョンアップを受け付けています。随時最新の内容でお届けします。ご希望のユーザーは62円切手6枚を希望メディア(3.5"または5")を明記した上で、弊社まで送ってください。(ブランクディスク2枚と返送用切手でも可)

■X680x0用外付大容量ハードディスク■

プログラム・音楽データ・画像データ...とハードディスクの足りない方にオススメ。フォーマット済のため、接続後にすぐ使用できます（パーティション分割する場合は、一旦領域解放し、再度領域を確保してください）。

1.0GB（Q）平均アクセスタイム10ms	定価¥168,000-のところ、¥128,000-
1.2GB（Q）平均アクセスタイム10ms	定価¥198,000-のところ、¥148,000-
1.8GB（Q）平均アクセスタイム10ms	定価¥248,000-のところ、¥178,000-
2.4GB（S）平均アクセスタイム8ms	定価¥348,000-のところ、¥238,000-
2.4GB（F/5"）平均アクセスタイム11.5ms	定価¥338,000-のところ、¥223,000-
3.4GB（S/5"）平均アクセスタイム11ms	定価¥398,000-のところ、¥288,000-
240MB（Q）平均アクセスタイム10ms	特別提供価格¥44,800-

QはQuantumドライブ使用　SはSeagateドライブ使用　FはFusitsuドライブ使用
(容量はすべてアンフォーマット状態ですのでフォーマット後の容量は多少変わりますのでご了承ください)
すべてケーブル付。
その他の容量も取り扱っていますので、お問い合わせください。

お買求め・お問い合せは...

弊社製品は直販のみの販売でSHOPではお求めになれません。詳しい購入方法や細かい仕様などの資料を用意しておりますので、郵便番号・住所(都道府県からお願いします)・氏名を明記の上、ハガキにてご請求ください(代金を直接送らないで下さい)。

毎日沢山の資料請求のハガキが届いておりますが、配達先不明で返送されてくるものがあります。難しい文字には読み仮名を付けていただけると助かります。

電話でのお問い合せも受け付けておりますが、業務の都合により留守電に繋がることも御座いますのでご了承下さい。

NEO COMPUTER SYSTEMS™

株式会社ネオコンピュータシステム
120　東京都足立区綾瀬1-33-7-103
TEL　03-5680-7531(月曜から金曜AM10:00-PM4:00)
FAX　03-5680-7539
NET　03-5680-7533,03-5680-7534(INS-C)

シャープパソコン・周辺機器特価セール

X1シリーズ

型番	品名	標準価格	特別価格
CZ-300C	3.5"FDDx2(Compact Type)	¥388,000	~~¥285,000~~
CZ-310C	3.5"FDDx2:80MBHDD(Compact Type)	¥478,000	~~¥350,000~~
CZ-500C	5"FDDx2	¥398,000	~~¥289,000~~
CZ-510C	5"FDDx2：80MBHDD	¥488,000	~~¥360,000~~
CZ-820C	X1G MODEL10		¥16,800
CZ-822C	X1G MODEL30	¥118,000	¥39,800
CZ-830C	X1 twin	¥99,800	¥35,000
●周辺機器●			
AN-1506	15ピン6ピンディスプレイ変換ケーブル		¥1,700
AN-1508	15ピン8ピンディスプレイ変換ケーブル		¥1,700
CZ-300F	X1 3"フロッピーディスクドライブ	¥79,800	¥5,000
CZ-31F	X1 300F用増設ドライブ	¥59,800	¥3,000
CZ-501H	X1 増設用ハードディスクユニット	¥258,000	¥60,000
CZ-82F	X1 CZ-802C用増設ドライブ	¥59,800	¥6,000
CZ-8BF1	X1 フロッピーディスク I/F 5"2HD	¥14,800	¥11,500
CZ-8BGR2	X1 グラフィックボード	¥14,800	¥3,000
CZ-8BK2	X1 漢字ROM	¥19,800	¥16,800
CZ-8BO1	X1 フロッピーディスク I/F 5"2D	¥14,800	¥8,000
CZ-8EB3	X1 拡張I/Oボックス	¥33,800	¥28,000
CZ-8LM1	RS-232Cケーブル(平行)	¥7,200	¥6,000
CZ-8LM2	RS-232Cケーブル(クロス)	¥7,200	¥6,000

MZ・AXシリーズ

型番	品名	標準価格	特別価格
●ディスプレイ●			
MZ-1D10	12"モノクロディスプレイ	¥41,800	¥25,000
MZ-1D17	15"CRT(MZ-5500/6500)	¥124,000	¥49,800
●周辺機器●			
IP-1243	パソコンファクス25	¥30,000	¥8,000
MZ-1C05	3500用RS-232Cケーブル		¥3,340

型番	品名	標準価格	特別価格
MZ-1C17			¥2,000
MZ-1C18	700用I/Oケーブル		¥3,040
MZ-1C24	1P04用プリンタケーブル		¥7,200
MZ-1C25	700用プリンタケーブル		¥6,000
MZ-1C26	700用プリンタケーブル		¥6,240
MZ-1C32A	5500/6500用プリンタケーブル	¥7,800	¥5,440
MZ-1C40	RS-232Cケーブル		¥6,500
MZ-1E01	MZ-3500用RS-232Cボード	¥28,000	¥13,000
MZ-1E04	MZ-2000用プリンタI/F	¥10,000	¥6,000
MZ-1E08	MZ-2000/2200/80B用プリンタI/F	¥9,000	¥8,000
MZ-1E14	MZ-1500用クイックディスクI/F	¥9,800	¥3,000
MZ-1E18	MZ-2000用クイックディスクI/F	¥9,800	¥3,000
MZ-1E21	MZ-5500用GP I/F	¥36,000	¥12,000
MZ-1E22	MZ-5500用GPIB I/F	¥72,800	¥25,000
MZ-1E29	RS-232C I/F 300BT	¥17,800	¥9,800
MZ-1E32	MZ-2500用パラレルI/F	¥30,000	¥27,000
MZ-1E33	MZ-6500用パラレルI/F	¥34,800	¥28,000
MZ-1E39	MZ-2800用RS-232C I/F	¥39,800	¥13,000
MZ-1E44	MZ-6500用S-RN I/F	¥50,000	¥15,000
MZ-1E45	MZ-6500用RS-232C I/F	¥50,000	¥15,000
MZ-1M01	MZ-2000/2200用 16Bitボード	¥78,000	¥3,000
MZ-1M03	MZ-5500用数値演算プロセッサ	¥69,000	¥38,500
MZ-1M09	MZ-6500用8082-2演算プロセッサ	¥82,000	¥30,000
MZ-1M12	MZ-2861/6500 80287数値演算プロセッサ	¥90,000	¥45,000
MZ-1P06	ドットプリンタ	¥234,000	¥45,000
MZ-1P10A	24ドット80桁漢字プリンタ	¥245,000	¥75,000
MZ-1P22	熱転写漢字プリンタ	¥59,800	¥16,000
MZ-1P27	漢字水平プリンタ	¥268,000	¥75,000
MZ-1R01	MZ-2000/2200Gボード	¥39,800	¥10,000
MZ-1R06	MZ-5500用増設RAM	¥45,000	¥8,000
MZ-1R09	MZ-5500 VRAM	¥35,000	¥15,000
MZ-1R10	MZ-5500漢字ROM付	¥30,000	¥9,800
MZ-1R11	MZ-5500用増設256KBRAM	¥80,000	¥35,000
MZ-1R12	MZ-80B/2000/1500/700用RAM	¥35,000	¥8,000
MZ-1R14	MZ-5500用辞書ROM	¥40,000	¥22,000

型番	品名	標準価格	特別価格
MZ-1R16	MZ-5500用増設128KBRAM	¥30,000	¥8,000
MZ-1R21	MZ-1P10第二水準漢字ROM	¥38,000	¥13,000
MZ-1R24	MZ-1500用辞書ROM	¥22,000	¥6,000
MZ-1R26代	MZ-2500用増設RAM		¥10,000
MZ-1R35	MZ-2800用1MBRAM		¥19,000
MZ-1S13	MZ-1D17用チルトスタンド	¥12,000	¥5,000
MZ-1T02	MZ-2200用テープレコーダー	¥19,800	¥8,500
MZ-1T03	MZ-5500用テープレコーダー	¥12,000	¥8,500
MZ-1U09代	MZ-2500用拡張ボード		¥4,000
MZ-1V01	パソコン・プリンタ・コピー・ファクス	¥278,000	¥75,000
MZ-1X22	MZ用モデムユニット	¥21,800	¥13,000
MZ-1X30	MZ用1200/300モデムホン	¥98,000	¥19,800
MZ-2Z023	MZ-5500GW-BASIC		¥30,000
MZ-2Z029	MZ-6500TODAY		¥20,000
MZ-4Z001	MZ-6500/5500IBM-FORMAT CONVERSION		¥8,000
MZ-5Z013	MZ-1500クイックディスク通信ソフト		¥3,500
MZ-6F03	クイックディスク	¥450	¥400
MZ-6P06	MZ-1P06用トラクタフィーダ	¥15,000	¥7,500
MZ-6P16	MZ-1P22リボンカセット(黒)	¥1,500	¥1,000
MZ-6P17	MZ-1P22リボンカセット(カラー)	¥1,500	¥1,000
MZ-6P18	MZ-1P18/28用カットシートフィーダ	¥60,000	¥35,000
MZ-6P20	MZ-1P22/17用ロールホルダー	¥3,100	¥2,700
MZ-6P21	MZ-1P18/28/29黒リボンカセット	¥1,800	¥1,600
MZ-6P27	MZ-1P27用カットシートフィーダ	¥58,000	¥39,800
MZ-6P29	MZ-1P29用カットシートフィーダ	¥50,000	¥37,500
MZ-6Z25	MZ-5500用ストリーマユーティリティZプロセッサ	¥39,800	¥15,000
MZ-80P4B	136桁ドットプリンタ		¥48,000
MZ-8B104	MZ-2000/2200用GPIB I/F	¥45,000	¥18,000
MZ-8BC01	MZ-2000/2200用GPIBケーブル	¥18,000	¥8,000
MZ-8BGK	MZ-80用BGRAM2	¥39,000	¥10,000
SS-SC28M	MZ-2800ハンディコピーキット	¥49,800	¥10,000
UE-01	AX ICカードインターフェイス	¥45,000	¥30,000
UE-1E02	AX ICカードインターフェイス	¥45,000	¥30,000
UE-1R03	AX 1M増設RAMボード	¥100,000	¥65,000
UE-1R07	AX 辞書ROMボード	¥32,800	¥26,200

型番	品名	標準価格	特別価格
UE-1R09	AX 1M増設RAMボード	¥75,000	¥55,000
UE-1R11	AX 1M増設RAMボード		
UE-1R13	AX 辞書ROMボード	¥32,800	¥25,000
UE-1U01	AX スロットボックス	¥5,000	¥4,000
●ソフト●			
IP-1215	MZ-2500 COBOL		¥11,700
IP-1251	MZ-2800 デスクUP	¥88,000	¥10,000
IP-1253	MZ-2800 クリッパー	¥77,000	¥10,000
IP-1254	MZ-2800 プランUP	¥66,000	¥10,000
MZ-25ゲーム	DANGER BOX		¥1,000
MZ-25ゲーム	九玉伝		¥1,000
MZ-25ゲーム	ガレイドスコープ		¥1,000
MZ-25ゲーム	トリトーン		¥1,000
MZ-25ゲーム	ブラックオニキス		¥1,000
MZ-25ゲーム	ムーンチャイルド		¥1,000
MZ-25ゲーム	リザード		¥1,000
MZ-2Z012	MZ-5500付属ソフト		¥5,000
MZ-2Z016	MZ-5500付属ソフト		¥5,000
MZ-2Z023	MZ-5500 GW BASIC	¥50,000	¥30,000
MZ-2Z029	MZ-6500 TODAY	¥68,000	¥20,000
MZ-2Z065	MZ-6500 書院日本語ワープロ	¥69,800	¥28,000
MZ-4Z001	MZ-5500 IBM変換	¥30,000	¥8,000
MZ-6Z22	MZ-6500用CP/M 86BASIC	¥10,000	¥6,000
MZ-80T20A	MZ-80用マシン語	¥6,000	¥5,000
MZ-80T40A	MZ-80用 PASCAL(言語)	¥10,000	¥5,000
MZ-80T70A	MZ-80用 FDOS(OS)	¥20,000	¥7,000
MZ-80TU	MZ-80用システムプログラム	¥20,000	¥8,000
MZ-80TUB	MZ-80用バックアップツール	¥20,000	¥8,000
SUPER DEVICE MONITER "T"			¥11,200
スーパー修理屋さん			¥10,500

他にパソコン・ポケコン・周辺機器
大量在庫有り。
お問い合せ下さい。

〈全商品新品完全保証付〉
★シャープ・シャープ周辺機器(拡張機器全機種、プリンター他)・富士通・NEC取り扱い。
★シャープ・カシオポケコン全機種取り扱い。PACIFIC・YHP・キャノンも取り扱い。
★上記商品価格には、消費税は含まれておりません。

通信販売のお問い合せ、御注文は
TEL.0426-45-3001(本店)　FAX.0426-44-6002
●営業時間/10:00〜19:00●電話受付/9:00〜21:00 迄可●定休日/水曜日
SHARP SUPER EXE SHOP
アイビット電子株式会社　〒192　東京都八王子市北野町560-5

上記の広告商品は店頭販売もしております。

全国通信販売　北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店　（普）1752505

5年保証5年保証5年保証5年保証5年保証5年保証5年保証5年保証5年保証5年保証

新品パソコン業界最長5年保証

全国通販 簡単申込!! 代金引換OK

●店頭販売もしております。御来店大歓迎。
●単品販売もしております。詳しくは、お電話にてお問合せ下さい。
●HARD & SOFT。パソコン専門販売!!

●営業時間:AM10:00～PM7:00 日・祭:AM10:00～PM6:00 ●定休日:毎週水曜日・第3日曜日

お問い合わせ お申し込みは（着信人払い） フリーダイヤル **0120-01-4865**

お問合せ・お見積 **03-3655-4454** FAX **03-3655-4436**

《業界No.1のメンテナンスサポート》

最高の保証システム

①業界最長の新品パソコン5年保証。(※モニター・プリンター3年間保証!! ※一部商品は除きます。) ②中古パソコンの1年間保証(※モニター・プリンター6ヶ月間保証!!) ③初期不良交換期間3ヶ月(※新品商品に限らせていただきます。) ④永久買取保証。⑤配達日の指定OK!!(土曜・日曜・祭日もOK) ⑥夜間配達OK!!(※PM6:00～PM8:00の間。※一部地域は除きます。)

便利でお得な支払いシステム

①翌月一括払い手数料無料(ご利用下さい)。②業界No.1の低金利!! ③月々の支払いは¥1,000より。④9ヶ月先からのスキップ払いOK!! ⑤84回までの分割、ボーナス併用OK!! ⑥カレッジクレジット。⑦ステップアップクレジット。⑧ボーナスだけで10回払いOK!! ⑨現金一括支払いOK!! ⑩商品到着払いOK!!(代引き手数料が必要になります。)(10万円まで900円) ※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。
※店頭即決クレジットにて持ち帰りOK!

注目!!平成6年3月末一括払い、手数料(金利)無料。(平成6年1月末/2月末/3月末のいずれかをご指定ください。)

パソコンSHOP P-メディア オープン記念セール!

X68030ハードディスクセット

①
120M 外付

●CZ-500C(本体)
●CZ-608D(モニター)
●SHD-FMX120 (ロジテック)(ターミネーター、ケーブル付)
●フロッピーアタッシュケース(X68ロゴ入り)

合計定価¥492,800⬇
PM特価¥375,000

②
240M 外付

●CZ-500C(本体)
●CZ-608D(モニター)
●SHD-FMX240 (ロジテック)(ターミネーター、ケーブル付)
●フロッピーアタッシュケース

合計定価¥630,800⬇
PM特価¥389,000

③
80M 内蔵

●CZ-500C(本体)
●CZ-608D(モニター)
●CZ-5H08(シャープ)
●フロッピーアタッシュケープ

合計定価¥590,800⬇
PM特価¥399,000

④
160M 内蔵

●CZ-500C(本体)
●CZ-608D(モニター)
●CZ-5H16(シャープ)
●フロッピーアタッシュケース

合計定価¥627,800⬇
PM特価¥427,000

本体の変更の場合

①CZ-510Cに変更の場合¥71,000
②CZ-300Cに変更の場合¥ 1,000
③CZ-310Cに変更の場合¥64,000
上記加算して下さい。

モニターの変更の場合

①CZ-607D(チューナー付)に変更の場合¥ 3,000
②CZ-614D(チューナー付)に変更の場合¥31,000
③CZ-621D(B)………… に変更の場合¥58,000
加算して下さい。
※300シリーズにチューナー付のモニターを接続の場合CRTケーブルを購入して下さい。

X68000 Compact XVI セット

①
●CZ-674C(本体)
●CZ-608D(モニター)
●CZ-8NJ2(サイバーステック)

合計定価¥416,600⬇
PM特価¥169,000

②
●CZ-674C(本体)
●CZ-608D(モニター)
●5″FDD(5″×2)(CZ-6FD5同等品)
●CZ-8NJ2(サイバーステック)

合計定価¥456,400⬇
PM特価¥209,000

③
●CZ-674C(本体)
●CZ-614D(ケーブル付)
●CZ-8NJ2(サイバーステック)

合計定価¥466,800⬇
PM特価¥206,000

④
●CZ-674C(本体)
●CZ-614D(ケーブル付)
●5″FDD(5″×2)(CZ-6FD5同等品)
●CZ-8NJ2(サイバーステック)

合計定価¥496,600⬇
PM特価¥246,000

■モニターの変更の場合
CZ-607D-TN(定価¥ 99,800)に変更の場合¥ 3,000
CZ-621D(B) (定価¥168,000)に変更の場合¥58,000
加算して下さい。

X68000/X68030周辺機器

48ドット熱転写カラー漢字プリンタ

定価¥96,800▼
特価¥38,000

カラーイメージスキャナ

■JX-325X
定価¥190,000▼
特価¥143,000

FAX/DATAモデム(マイクロコア)

■MC14400FX
定価¥46,800▼
特価¥34,500

●CZ-6FD5……………定価¥ 99,800➡特価¥ 49,800
●CS-M120WA(コパル)…定価¥178,000➡特価¥119,000
●LMO-FMX330…………定価¥178,000➡特価¥135,000
●JX-220X……………定価¥168,000➡特価¥ 89,800
●CZ-6TU……………定価¥ 33,100➡特価¥ 23,900
●SH-5BE4-8M(I/O)……………………特価¥ 45,500
●2MB増設RAMボード(I/O)………………特価¥ 23,000
●4MB増設RAMボード(I/O)………………特価¥ 38,300
●CZ-6BE2A……………定価¥ 59,800➡特価¥ 42,800
●CZ-6BE2B……………定価¥ 54,800➡特価¥ 39,300
●CZ-6BE2D……………定価¥ 54,800➡特価¥ 39,300
●CZ-6BM1A……………定価¥26,800➡特価¥19,300
●CZ-6BV1……………定価¥21,000➡特価¥15,200
●CZ-6BS1……………定価¥29,800➡特価¥21,500
●CZ-5MPI……………定価¥54,800➡特価¥42,000
●SX-68MII(MIDI)(サコム)定価¥19,800➡特価¥13,500
●SX-68SC(SCSI)(〃)定価¥26,800➡特価¥17,500
●CZ-5BE4……………定価¥54,800➡特価¥42,000
●CZ-5ME4……………定価¥49,800➡特価¥38,000
●MD-24XT10V(オムロン)‥定価¥29,800➡特価¥22,500
●MD-96XT10V(〃)‥定価¥46,800➡特価¥32,000

■X68000/68030ソフト

●Z's STAFF PRO68K Ver.3.0(ツアイト) ………定価¥58,000➡特価¥37,500
●Z's TRIPHONY デジタルクラフト(ツアイト) ………定価¥39,800➡特価¥27,000
●マジックパレット(ミュージカルプラン) ………定価¥19,800➡特価¥14,200
●たーみのる2(SPS) ………定価¥17,800➡特価¥13,000
●Mu-1 Super(サンワード) ………定価¥39,800➡特価¥28,500
●マチエール Ver.2.0 ………定価¥39,800➡特価¥28,800
●CZ-225BSV Multiword Ver.2.0 ………定価¥32,000➡特価¥23,000
●CZ-247MSD MUSIC PRO68K(MIDI) ………定価¥28,800➡特価¥20,500
●CZ-261MSD MUSIC studio PRO68K Ver.2.0 ………定価¥28,800➡特価¥21,200
●CZ-263GWD Easypaint SX-68K ………定価¥12,800➡特価¥ 9,800
●CZ-264GWD Easydraw SX-68K ………定価¥19,800➡特価¥15,300
●CZ-265HSD New Print Shop Ver.2.0 ………定価¥20,000➡特価¥15,400
●CZ-272CWD Communication SX68K ………定価¥19,800➡特価¥14,500
●CZ-275MWD SOUND SX68K ………定価¥15,800➡特価¥11,500
●CZ-288L WD開発キット(workroom) ………定価¥39,800➡特価¥29,700
●CZ-290TWD SX-WINDOWディスクアクセサリー集 ………定価¥14,800➡特価¥11,500
●CZ-294SS(5″)/SSC(3.5″)SX-WINDOW Ver.3.0 ………定価¥19,800➡特価¥15,200
●CZ-296LSD C-Compiler PRO68K Ver.2.1 NEW KIT ………定価¥44,800➡特価¥32,500

SHARP 液晶ペンコム PI-3000

定価¥65,000
特価TEL下さい。

●PC-E650
定価¥33,000➡特価¥25,500
●PC-E200
定価¥22,000➡特価¥17,000
●PC-1262
定価¥24,800➡特価¥19,000

特選パソコンラック&OAチェアー
(消費税込み)(送料無料、離島を除く)

4段¥9,785

●ラック&チェアーセット
●布張り(ダークグレー)
●ガスシリンダー
(4段) ¥18,540

■通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)
〔銀行振込でお申し込みの方〕●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。(電信扱いでお振込み下さい。)
〔クレジットでお申し込みの方〕●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1,000円以上。

〔振込先〕 さくら銀行 新小岩支店 普通預金 3384331 (有)ピーメディア

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.9	3.9	4.9	5.4	8.4	11.4	15.9	20.9	26.9	34.9

有限会社 ピーメディア
〒124 東京都葛飾区新小岩2丁目2番地20号
FAX.03-3655-4436
お問い合わせ お申し込みは フリーダイヤル 0120-01-4865

●本広告の掲載の商品の価格については、消費税は含まれておりません。※掲載の価格は、店頭と異なる場合がございます。※価格は変動することがございますので、最新の価格は、お電話にてお問い合わせください。

満開の電子(でんこ)ちゃん

作・え 岡村祭(おかむらまつり)

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。

★定期購読の場合＝購読料６ヶ月分6,000円（送料サービス、消費税込）を、
現金書留または郵便振替で下記の宛先へお送り下さい。

現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
郵便振替の場合：東京 5－362847 ㈱満開製作所

- ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
- 3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
- 新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
- 製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。

★TAKERUでお求めの場合＝１部につき1,200円（消費税込）です。

- 定期購読版と内容が一部異なる場合があります。御了承下さい。
- お問い合わせ先 TEL(03)3554－9282（月～金 午前11時～午後６時）

（なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります）

木下 達也（愛媛県）

電脳倶楽部の存在を知りつつも、不安だとか怪しいとかいった理由で購読に踏み切れない方がいるようですね。

そんな方に、ひとこと。

「とりあえず、試しに6ヶ月定期購読してはいかがですか。」

便利なツール、楽しいゲーム、役立つデータ、読み物、グラフィック、ビープ音など、電脳倶楽部はまさにX68000を楽しみたいあなたのための雑誌といえるでしょう。

食わず嫌いはいけません。まずはこの世界を味わってみてください。

ツクモパソコン本店II 3F、OAシステムプラザ（札幌・仙台・横浜西口・大須・大阪日本橋・岡山・広島・福岡・鹿児島）でも販売中／

宍戸留美

夢は、い・た・だ・きよ……

コンパクトXVI改造機。弊社にて1年保証。クロックは10／16／24の3モード。16／24MHzは背面トグルスイッチにより切替。RED ZONEの24MHzでは正常動作しないソフト等がありますが、10／16MHzでご使用になれます。

X68000 Compact XVI改
RED ZONE ￥160,000

・シャープ製CZ-6FD5完全コンパチブル・オートイジェクト機能付・ドライブ番号切替スイッチ付・木製（ナラ材）フロントパネル・対応機種／CZ-674C／300C／310C／500C／510C
・カラーリングオプションはプラス5,000円です。

シャープ製CZ-6FD5完全コンパチFDD（MK-FD1）

満開式軟盤駆動装置壱號
￥39,800（税別、カラーモデル￥44,800）

Compact XVI改　5インチFDD
RED ZONE + MK-FD1
セット価格 **￥180,000**（税別）
（FDカラーモデルは＋￥5,000）

新登場　満開式硬盤駆動装置弐號
1GバイトSCSIハードディスクユニット
MK-HD2
￥150,000→**￥125,000**（税別）
平均アクセスタイム8.3msの高速HDがこの価格
直販のみです。

98バスマウスを68で使っちゃうアダプタ
MOUSEJACK68-98
上記パソコンショップでもお求めになれます。
MK-MJ1 ￥4,000（税別）

当ショップは通販専門店です。X680×0用各種ハード・ソフトも取り扱っております。お電話にて商品リストと注文書をご請求ください。RED ZONEのご購入には承諾書が必要です。合わせてご請求ください。

〒171 東京都豊島区長崎1−28−23Muse西池袋2F
TEL (03)3554−7441　FAX (03)3554−3856

パソコンショップ満開
㈱満開製作所

T1002179010605　雑誌 02179-1